Humboldt University of Berlin
Berlin School of Business and Economics
Chair of Statistics

Dynamic prediction in flexible Bayesian additive joint models

Author:
Jil Kollmus-Heege
588495

First Examiner:
Prof. Dr. Sonja Greven

Second Examiner:
Prof. Dr. Nadja Klein

A thesis submitted in partial fulfillment of the requirements
for the degree of *Master of Science in Statistics*

July 16, 2020

Abstract

Medical treatments tailored to the individual patient commonly known as personalized medicine have received increasing attention over the past decades. A promising method for analysis and prediction in the field of personalized medicine are joint models. Joint models enable to *jointly* model longitudinal and time to event data based on which individualized dynamic predictions can be derived. In this work we extended the existing framework of flexible Bayesian additive joint models by implementing a dynamic prediction for the time to event and longitudinal outcomes. Flexible Bayesian additive joint models are a specific class of joint models which allow for additional subject-specific flexibility. We investigate if this additional flexibility improves the overall predictive quality by comparing our framework to a standard joint modeling approach by means of accuracy measures explicitly developed for the joint modeling framework. Our results show that a dynamic prediction based on flexible Bayesian additive joint models generally outperforms the standard model approach. Moreover, we could show that if the true underlying association of longitudinal marker and event is nonlinear the dynamic prediction generally performs better when modeling a time-varying association instead of a constant one.

Contents

1 Introduction

Individualizing a patient's treatment plan, known as personalized medicine, has received great attention over the past decades. Studies have shown that the way individuals respond to a medication highly depends on their ethnic background and specific genes, such that the medication might be helpful for one patient but harmful or even leading to death for another patient (Mukherjee and Topol, 2002; Currie et al., 2006). This is one reason why adverse drug reaction is the fifth highest cause of death in the US (Mukherjee and Topol, 2002). Recent efforts in biostatistics have lead to an increasing development of methods that take more into account this variability between individuals. One method that contributes to this branch of medicine are joint models for longitudinal and time-to-event data, based on which individual dynamic predictions can be derived. These models are well suited for personalized medicine since they use random effects and are thus subject-specific by nature (Rizopoulos et al., 2015).

Joint models are applicable in settings where the interest lies in analyzing the effect of a longitudinal measure, often a time-varying biomarker, on some event of interest which can be the onset of a disease or the time of death. A well known field of application is in AIDS research where the association between CD4 cell count and the time until seroconversion or death is examined (Pawitan and Self, 1993; Wulfsohn and Tsiatis, 1997). The main idea to unbiasedly estimate this association is to model the longitudinal observations in a submodel, usually using a linear mixed model, jointly with a survival submodel. This way of modeling the biomarker allows to account for a potential measurement error in the biomarker and the marker enters the survival submodel as a continuous variable in time although it is only observed intermittently. In a shared random effects approach unbiased estimates are obtained by deriving a joint likelihood assuming that the random effects influence both submodels, longitudinal and survival, and that the submodels are conditionally independent, given those random effects.

Based on a fitted joint model subject-specific dynamic predictions for either the survival outcome or the longitudinal marker value can be derived. Those predictions are beneficial since (1) they are individualized, (2) they can be updated as soon as new observations are collected and (3) they utilize the whole longitudinal history (Rizopoulos et al., 2014). With regard to personalized medicine, the aim of such predictions is that physicians can individually adjust the treatment plan by using predictions of a disease progression or the occurrence of an event and therefore are able to improve their patient's prospects. Rizopoulos (2011) for instance investigate the association between CD4 cell count and survival for patients with advanced HIV. Based on dynamic predictions for the survival outcome, they evaluate how suitable CD4 cell count discriminates between individuals that died within a medical relevant time frame after their last assessment and individuals that are still alive. Another more progressive work by Tomer et al. (2019) suggests to personalize the screening intervals for low-risk prostate cancer patients based on dynamic predictions in order to avoid unnecessary painful biopsies.

Many joint models only focus on modeling simple parametric longitudinal trajectories (Brown et al., 2005). However, a lot of trajectories are highly nonlinear and/or differ between individuals which is at the core of personalized medicine. This can be seen for example in Figure 1 where the observed logarithm of serum bilirubin, a biomarker that is associated with the chronic, fatal liver disease primary biliary cirrhosis (PBC), is shown for five randomly selected subjects from the pbc2 data, available in the `R` package `JMbayes`. In order to capture this non-linearity joint models have been developed that allow a more flexible modeling using for example a spline-based approach (Ding and Wang, 2008; Brown et al., 2005; Rizopoulos and Ghosh, 2011). Yet, the disadvantage of such an approach is specifying an appropriate number and position of knots since a smaller number of knots tends to underfit the trajectory and a larger number may result in wiggly functions. To overcome

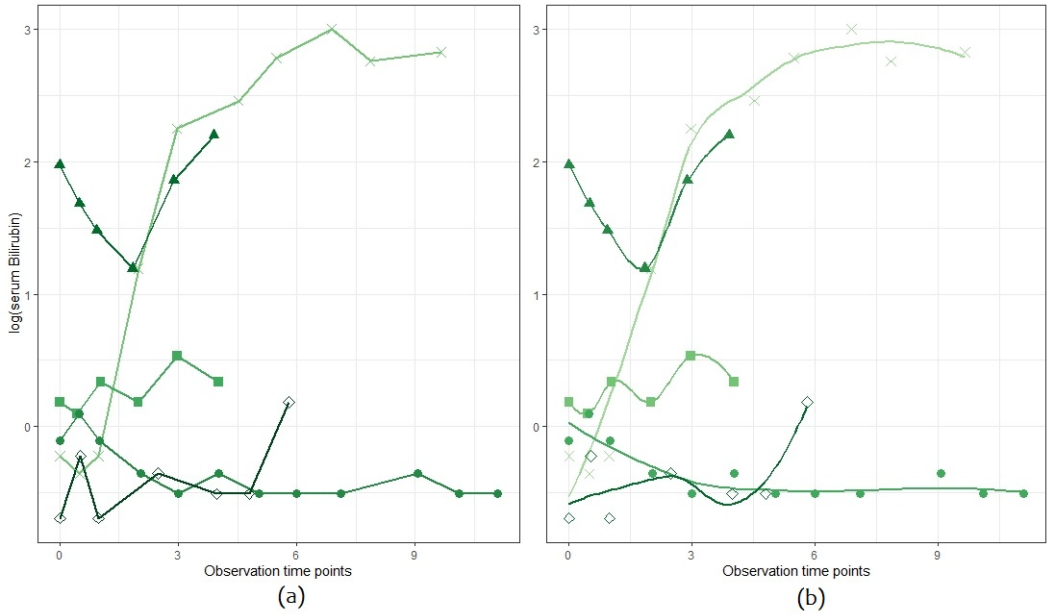

Figure 1: Longitudinal marker values of log(serum Billirubin) for five randomly selected subjects from the data pbc2: observed values (points) and (a) linear interpolation (lines); (b) LOESS smoother (lines)

these problems Köhler et al. (2017) proposed a class of joint models, so called flexible Bayesian additive joint models, that model the longitudinal trajectory using P-splines which are based on a B-splines basis but a penalty applied to the corresponding coefficients avoids selecting an explicit number of knots. Further, a time-varying association between the biomarker and time to event can be modeled. All parameters in this class of models are estimated in a Bayesian approach since the estimation in a frequentist approach requires integrating over the potentially high-dimensional random effects distribution (Köhler et al., 2017).

The main focus of this work is to implement a dynamic prediction that is based on flexible Bayesian additive joint models and then to evaluate the predictive quality. This is of special interest since Köhler et al. (2017) have shown in a simulation study that more flexible joint models frequently outperform less flexible joint models when comparing the bias and MSE of the coefficient estimates. This is especially the case for the fit of the longitudinal trajectories since less flexible joint models seem to underestimate the nonlinearity in the trajectory (Köhler et al., 2017). Moreover, Andrinopoulou et al. (2018) showed in a simulation study that individualized dynamic predictions are overall improved when assuming a time-varying association between the marker and the time to event process, even if the true underlying association is only constant.

Therefore, in this work, we are going to evaluate if a more flexible joint model setup and also a time-varying association may improve the accuracy of subject-specific dynamic predictions by implementing the dynamic prediction introduced by Rizopoulos (2011) and Proust-Lima and Taylor (2009) for the class of flexible Bayesian additive joint models. In order to assess the quality of our dynamic prediction, we are going to conduct a simulation study in which our predictions are going to be compared to the ones obtained from a less flexible class of joint models. Therefore, both dynamic predictions are compared by means of evaluation methods popular in survival analysis. Moreover,

we are going to evaluate and demonstrate the main usage of the implemented prediction based on a reanalysis of the aforementioned PBC data that is widely used in the joint modeling framework. In the analysis we are further going to compare our results again to the less flexible joint model class.

The remainder of this work is structured as follows: First, we are going to give an introduction to longitudinal data analysis as well as time to event data analysis which form the basic methodology for joint models. Then, in Section 2.3, joint models for longitudinal and time to event data, which we are mostly going to denote as only joint models, are introduced. In Section 3, we will extend the standard joint model to the aforementioned flexible Bayesian additive joint model. In the next Section 4, we are first going to derive the dynamic prediction for both, the survival and longitudinal outcome, then present important evaluation measures from time to event data analysis which were adapted to our dynamic setting, and third give details on how this prediction is implemented for flexible Bayesian additive joint models. In Section 5 we will reanalyse the PBC data and evaluate the dynamic prediction and thereby demonstrate its main usage. In order to evaluate the predictive quality, we conducted a simulation study whose design and results are shown in Section 6.

2 Joint models for longitudinal and time to event data

2.1 Longitudinal data analysis

In many studies and especially in clinical trials it is common to collect repeated measurements of a certain variable, e.g. a bio marker, over time. The main advantage of analyzing longitudinal data, compared to only cross-sectional data, is that they take into account two types of information. First, the cross-sectional information reflecting differences between individuals. And second, the time series information reflecting changes within individuals over time. To analyze this type of data we cannot use standard statistical tools like linear regression since the observations within one subject are potentially correlated and therefore the assumption of independent observations is violated. A popular model that takes this correlation into account is the linear mixed model: Let $\boldsymbol{y}_i = [y_{i1}, y_{i2}, ..., y_{in_i}]^\top$ be the n_i dimensional response vector holding all longitudinal measurements on subject i, $i = 1, 2, ..., n$, where y_{ij} denotes the observed value for subject i at time point t_{ij}, $j = 1, ..., n_i$. The linear mixed effects model (Verbeke and Molenberghs, 2000) can be written as

$$\begin{cases} \boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{b}_i + \boldsymbol{\varepsilon}_i, \\ \boldsymbol{b}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{D}), \\ \boldsymbol{\varepsilon}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_i), \\ \boldsymbol{b}_1, ..., \boldsymbol{b}_n, \boldsymbol{\varepsilon}_1, ..., \boldsymbol{\varepsilon}_n \text{ independent}, \end{cases} \tag{2.1}$$

where $\boldsymbol{X}_i$ and $\boldsymbol{Z}_i$ are known $n_i \times p$ and $n_i \times q$ design matrices, $\boldsymbol{\beta}$ is a p dimensional vector of fixed effects, $\boldsymbol{b}_i$ is a q dimensional vector of subject-specific random effects and $\boldsymbol{\varepsilon}_i$ is a vector containing the residuals ε_{ij}, $j = 1, 2, ..., n_i$. Furthermore, $\boldsymbol{D}$ is a $q \times q$ dimensional covariance matrix and $\boldsymbol{R}_i$ being the $n_i \times n_i$ covariance matrix of the residuals. The random effects and residuals are assumed to be independent. For a compact version we define: the response vector $\boldsymbol{y} = [\boldsymbol{y}_1^\top, ..., \boldsymbol{y}_N^\top]^\top$, the design matrices $\boldsymbol{X} = [\boldsymbol{X}_1^\top, ..., \boldsymbol{X}_n^\top]^\top$ and $\boldsymbol{Z} = \text{diag}(\boldsymbol{Z}_1, ..., \boldsymbol{Z}_n)$, the random effects $\boldsymbol{b} = [\boldsymbol{b}_1^\top, ..., \boldsymbol{b}_n^\top]^\top$ as well as the block diagonal covariance matrices $\boldsymbol{G} = \text{diag}(\boldsymbol{D}, ..., \boldsymbol{D})$, with n times matrix $\boldsymbol{D}$ on its diagonal, and $\boldsymbol{R} = \text{diag}(\boldsymbol{R_1}, ..., \boldsymbol{R_n})$ for the random effects and the residuals, respectively.

Model (2.1) can be reformulated as a two stage, hierarchical model (Fahrmeir et al., 2009), where the *conditional* distribution of $\boldsymbol{y}$ can be derived as

$$\boldsymbol{y} \mid \boldsymbol{b} \sim \mathcal{N}(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{b}, \boldsymbol{R}). \tag{2.2}$$

In this representation the expectation of $\boldsymbol{y}$ is a function of the fixed and random effects. The *marginal* distribution can be derived as

$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{V}), \quad \boldsymbol{V} = \boldsymbol{R} + \boldsymbol{Z}\boldsymbol{G}\boldsymbol{Z}^\top \tag{2.3}$$

where the expectation of $\boldsymbol{y}$ depends now only on the fixed effects but the random effects add an extra term to the covariance.

To give an overview of the estimation of the fixed and random effects, we for now assume the variance $\boldsymbol{V}$ of the marginal representation (2.3) to be known. Thus, estimates of the fixed effects can be obtained using generalized least squares (GLS). This yields the estimator $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{V}^{-1} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{V}^{-1} \boldsymbol{y}$ which is the best linear unbiased estimator (BLUE) for $\boldsymbol{\beta}$ by Gauss-Markov theorem. The predictions for the random effects $\boldsymbol{b}$ can be obtained from the best linear predictor (BLP) $\hat{\boldsymbol{b}} = \boldsymbol{G}\boldsymbol{Z}^\top \boldsymbol{V}^{-1}(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})$. In fact, these estimators coincide with the ones resulting from maximum likelihood estimation (MLE): due to $p(\boldsymbol{y}, \boldsymbol{b}) = p(\boldsymbol{y} \mid \boldsymbol{b})p(\boldsymbol{b})$, the log likelihood can be derived, omitting additive constants, as

$$l(\boldsymbol{\beta}, \boldsymbol{b}) \propto -\frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\boldsymbol{b})^\top \boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\boldsymbol{b}) - \frac{1}{2}\boldsymbol{b}^\top \boldsymbol{G}^{-1}\boldsymbol{b}. \tag{2.4}$$

Maximizing this expression coincides with the minimization of a *penalized* least squares criteria that for example is of following form

$$\min_{\boldsymbol{\beta}, \boldsymbol{b}} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\boldsymbol{b})^\top \boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\boldsymbol{b}) + \lambda \boldsymbol{b}^\top \boldsymbol{G}^{-1}\boldsymbol{b}, \tag{2.5}$$

where λ is called the smoothing parameter that controls the amount of penalization. Hence, the second term in the log likelihood (2.4) takes into account the fact that $\boldsymbol{b}$ stems from a distribution with mean $\boldsymbol{0}$ and covariance matrix $\boldsymbol{G}$ and penalizes violations from $E(\boldsymbol{b}) = \boldsymbol{0}$ weighted by $\boldsymbol{G}$.

So far we assumed the variance $\boldsymbol{V}$ of model (2.3) to be known. However, to be able to obtain feasible estimates for $\boldsymbol{\beta}$ and $\boldsymbol{b}$ we need estimates for the variance components of $\boldsymbol{V}$. Therefore, let $\boldsymbol{G} =: \boldsymbol{G}(\boldsymbol{\vartheta})$ and $\boldsymbol{R} =: \boldsymbol{R}(\boldsymbol{\vartheta})$ and thus also $\boldsymbol{V} =: \boldsymbol{V}(\boldsymbol{\vartheta})$ depend on the unknown variance parameters $\boldsymbol{\vartheta}$. By maximizing the log likelihood of the marginal model

$$l(\boldsymbol{\beta}, \boldsymbol{\vartheta}) \propto -\frac{1}{2}\left\{\log|\boldsymbol{V}(\boldsymbol{\vartheta})| + (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top \boldsymbol{V}(\boldsymbol{\vartheta})^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})\right\} \tag{2.6}$$

for $\boldsymbol{\beta}$ and inserting the resulting estimate $\tilde{\boldsymbol{\beta}}(\boldsymbol{\vartheta}) = (\boldsymbol{X}^\top \boldsymbol{V}(\boldsymbol{\vartheta})^{-1} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{V}(\boldsymbol{\vartheta})^{-1} \boldsymbol{y}$ again in equation (2.6), we get the profile likelihood $l_p(\boldsymbol{\vartheta})$. Maximizing the profile likelihood with respect to $\boldsymbol{\vartheta}$ gives the ML-estimate $\hat{\boldsymbol{\vartheta}}_{ML}$. However, this estimate $\hat{\boldsymbol{\vartheta}}_{ML}$ is biased downwards since the loss of degrees of freedom due to the estimation of $\tilde{\boldsymbol{\beta}}$ is not taken into account. For this reason, the estimation of $\boldsymbol{\vartheta}$ is often based on the restricted likelihood $l_R(\boldsymbol{\vartheta})$ that can be obtained by integrating out $\boldsymbol{\beta}$ in the likelihood (2.6). Maximizing the restricted likelihood with respect to $\boldsymbol{\vartheta}$ gives the restricted ML-estimate $\hat{\boldsymbol{\vartheta}}_{REML}$. Plugging the resulting variance estimate $\boldsymbol{V}(\hat{\boldsymbol{\vartheta}}_{REML})$ into the estimator for the fixed effects $\hat{\boldsymbol{\beta}}$ and random effects $\hat{\boldsymbol{b}}$ yields feasible estimates for $\boldsymbol{\beta}$ and $\boldsymbol{b}$, respectively.

2.2 Time-to-event data analysis

In time-to-event data analysis, often also referred to as survival analysis, the interest lies in analyzing the time until a dichotomous event of interest such as the onset of a disease or death occurs. In this case the response variable is the time until this event. The analysis of such data is often complicated by *censoring* which for example means that a subject drops out of the study before the actual event

happened. In consequence, the true event time cannot be observed. Instead we only observe the censoring time which is the last time point where this subject was known to be event free. This concept is called *right censoring* since the event of interest is known to take place after the last observed time point. Other censoring mechanisms are *left censoring* where the event happens before the observed event time and *interval censoring* where the actual event occurs in the interval between two observed time points. When modeling time-to-event data it is important to take into account the potential censoring mechanism. Therefore, in the following we are going to present basic concepts and a popular model for estimating the time to a potentially right censored event.

2.2.1 Important concepts

Let T_i denote the observed event time for subject i, $i = 1, 2, ..., n$, which is equal to the true event time T_i^* if subject i is not censored. Otherwise T_i equals the censoring time C_i. In general, T_i is defined as the minimum of the true event time and the censoring time. Further, we introduce the event indicator $\delta_i = I(T_i^* \leq C_i)$ that takes 1 if the subject experiences the event. The survival function $S(t)$ is generally used to describe the distribution of T^* with

$$S(t) = \Pr(T^* > t) = \int_t^\infty f(t)\, dt \tag{2.7}$$

where $f(.)$ denotes the corresponding probability density function. Another important function is the hazard function $h(t)$ with

$$h(t) = \lim_{dt \to 0} \frac{\Pr(t \leq T^* < t + dt \mid T^* \geq t)}{dt}, \quad t > 0 \tag{2.8}$$

which describes the instantaneous rate at which an event occurs, given it did not occur until t. Note, that the hazard function completely specifies the distribution of T^* which means that we can derive the survival function and also the corresponding density from the hazard via

$$\begin{aligned} S(t) &= \exp\left[-\int_0^t h(u)\, du\right] \\ &= \exp\left[-\Lambda(t)\right], \end{aligned} \tag{2.9}$$

where $\Lambda(t) = \int_0^t h(u)\, du$ is called cumulative hazard function and

$$f(t) = h(t)\, S(t), \tag{2.10}$$

respectively. For a detailed derivation of the connection between the different functions see A.

2.2.2 Cox model

The Cox model (Cox, 1972) is a widely used and popular model to fit a time-to-event model. The hazard for one subject i, $i = 1, 2, ..., n$, in the Cox model specifies that

$$h_i(t) = h_0(t) \exp(\boldsymbol{w}_i^\top \boldsymbol{\gamma}), \tag{2.11}$$

where $h_0(t)$ is the baseline hazard that is an unknown, positive function and $\boldsymbol{\gamma}$ is a vector of unknown parameters that linearly link the observed covariates $\boldsymbol{w}_i$ to the log hazard. As the baseline hazard is assumed to be the same for all subjects, the risk differences between two subjects are completely

determined by its time-constant, linear predictors which can be seen in more detail by taking the hazard ratio of two individuals i and j, $i \neq j$, at time point t

$$\frac{h_i(t)}{h_j(t)} = \frac{h_0(t)\exp(\boldsymbol{w}_i^\top \boldsymbol{\gamma})}{h_0(t)\exp(\boldsymbol{w}_j^\top \boldsymbol{\gamma})}) = \exp\big((\boldsymbol{w}_i^\top - \boldsymbol{w}_j^\top)\boldsymbol{\gamma}\big). \tag{2.12}$$

Hence, the hazard ratio is time-constant and does not depend on the baseline hazard anymore leading to proportional hazard rates (Klein and Moeschberger, 2006).

For the derivation of the likelihood it is important to take into account the potentially right censored event times. Therefore, the likelihood consists of two parts. First, the non-censored subjects following the density $f_i(t)$ and second the censored subjects that contribute the probability $S_i(t) = \Pr(T_i^* > t)$ to the likelihood yielding

$$\begin{aligned} L(\boldsymbol{\gamma}) &= \prod_{i=1}^{n} f_i(T_i)^{\delta_i}\, S_i(T_i)^{1-\delta_i} \\ &= \prod_{i=1}^{n} h_i(T_i)^{\delta_i}\, S_i(T_i) \\ &= \prod_{i=1}^{n} h_i(T_i)^{\delta_i}\, \exp\left[-\int_0^t h_i(u)du\right] \end{aligned} \tag{2.13}$$

where the second equality follows from inserting the definition of the density $f_i(t) = h_i(t)\, S_i(t)$.

Given the property of proportional hazard rates, we can estimate the effect of the parameters $\boldsymbol{\gamma}$ on the hazard leaving the baseline hazard $h_0(t)$ completely unspecified. This is done by deriving a partial likelihood: Starting from (2.13) we can rewrite the likelihood as

$$L(\boldsymbol{\gamma}) = \prod_{i=1}^{n} \left(\frac{h_i(T_i)}{\sum_{j:T_j \geq T_i} h_j(T_i)}\right)^{\delta_i} \left(\sum\nolimits_{j:T_j \geq T_i} h_j(T_i)\right)^{\delta_i} S_i(T_i) \tag{2.14}$$

by multiplying and dividing the term $(\sum_{j:T_j \geq T_i} h_j(T_i))^{\delta_i}$. Cox (1972) argues that most information on $\boldsymbol{\gamma}$ is in the first term, while the loss of information from leaving out the last two terms is usually slight. Thus, he suggests using the partial likelihood

$$\begin{aligned} pl(\boldsymbol{\gamma}) &= \prod_{i=1}^{n} \left(\frac{h_i(T_i)}{\sum_{j:T_j \geq T_i} h_j(T_i)}\right)^{\delta_i} \\ &= \prod_{i=1}^{n} \left(\frac{\exp(\boldsymbol{w}_i^\top \boldsymbol{\gamma})}{\sum_{j:T_j \geq T_i} \exp(\boldsymbol{w}_j^\top \boldsymbol{\gamma})}\right)^{\delta_i} \end{aligned} \tag{2.15}$$

which does not depend on the baseline hazard anymore. Parameter estimates for the coefficient vector $\boldsymbol{\gamma}$ can then be obtained using ML on the partial likelihood which leads to consistent and asymptotically normally distributed estimates with mean $\boldsymbol{\gamma}$ being the true parameter vector (Tsiatis, 1981).

2.2.3 Extended Cox model

The previously introduced Cox model only allows for time-constant variables in the specification of the hazard function. In order to also include time-varying covariates Andersen and Gill (1982)

extended the existing Cox model to the *extended* Cox model which however is only unbiased for *external* time-varying covariates. Before presenting the model, let us introduce the concept of *external* and *internal* time-varying covariates by following Chapter 6 in Kalbfleisch and Prentice (2011). Let $x(t)$ denote a single, time-varying covariate that can be observed at time point t. External time-varying covariates are all variables which fulfill

$$\Pr\big(T^* \in [u,t) \mid x(u), T^* > u\big) = \Pr\big(T^* \in [u,t) \mid x(t), T^* > u\big) \tag{2.16}$$

for all u, t such that $0 < u \leq t$. This means that the probability for an event to occur in the interval $[u,t)$ should be independent from the fact that we already have observations for the time-varying variable in that interval. An equivalent definition is

$$f\big(x(t) \mid x(u), T^* > u\big) = f\big(x(t) \mid x(u), T^* = u\big), \quad t > u, \tag{2.17}$$

where $f(.)$ denotes the density of the value for the longitudinal process at a certain point in time. This representation formalizes the idea that the variable's future path is not affected by the occurrence of the event at some previous time point. On the other hand, internal time-varying covariates are all time-varying covariates that neither satisfy (2.16) nor (2.17). Typically, they arise as time-dependent measurements taken from the subjects under study, e.g. a biomarker. The exact distinction between those two types of variables is important since internal time-varying covariates require special treatment and would lead to biased estimates in the extended Cox model.

The extended Cox model then models the association between the external time-varying covariates and the hazard as

$$h_i(t) = h_0(t) \exp\left(\boldsymbol{w}_i^\top \boldsymbol{\gamma} + \boldsymbol{x}_i(t)^\top \boldsymbol{\beta}\right) \tag{2.18}$$

where $\boldsymbol{w}_i$ is a vector of time-constant baseline covariates and $\boldsymbol{x}_i(t)$ contains all time-varying covariates. Note, that in the case of time-dependent covariates we no longer have proportional hazard rates. Taking the hazard ratio of two subjects i and j, $i \neq j$, we get

$$\frac{h_i(t)}{h_j(t)} = \frac{h_0(t) \exp(\boldsymbol{w}_i^\top \boldsymbol{\gamma} + \boldsymbol{x}_i(t)^\top \boldsymbol{\beta})}{h_0(t) \exp(\boldsymbol{w}_j^\top \boldsymbol{\gamma} + \boldsymbol{x}_j(t)^\top \boldsymbol{\beta})} = \exp\left\{(\boldsymbol{w}_i - \boldsymbol{w}_j)^\top \boldsymbol{\gamma} + (\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t))^\top \boldsymbol{\beta}\right\}, \tag{2.19}$$

where the ratio now depends on time and thus is not time-constant but still independent of the baseline hazard.

This model already provides a simple approach to estimate the effect of a longitudinal biomarker on the hazard for an event. However, we would assume the marker to be observed without any measurement errors and to be constant between observations while their true underlying process is continuous in time. To overcome these issues, two-stage models have been developed. In these, the longitudinal observations are first modeled in a linear mixed model and the resulting predictions are then imputed into a survival model (Dafni and Tsiatis, 1998). However, these models still ignore the internal structure of longitudinal biomarkers which may lead to biased estimates of the effect towards zero (Prentice, 1982). In order to unbiasedly estimate the association of a longitudinal biomarker and a survival process joint models have been developed.

2.3 Joint models

Joint models are applicable in settings where subjects are observed over time and the aim is to determine the effect of a longitudinal covariate on the time to an event of interest. Such settings

are common when monitoring for example the progression of a disease. In this case the longitudinal covariate is typically a biomarker and the event of interest could be death or the onset of a disease. Since measurements of the biomarker are taken from the subject under study, it is very likely that they are influenced by the event which violates assumption (2.17) and thus they are considered internal. Further, they are generally measured with an error and their value is only known at specific time points. Therefore, the above introduced extended Cox model which models the association of a time-varying covariate and a time-to-event outcome results in biased estimates and standard errors (Prentice, 1982).

Joint models, on the other hand, provide unbiased estimates for this association by using a shared parameter approach where a latent parameter is assumed to influence both the longitudinal and time to event submodels and thus links those two models. A joint likelihood for both submodels can then be derived under the assumption of conditional independence, given the latent parameter. In the literature of joint models different models exist that differ in how they specify this latent parameter as well as the association of the longitudinal marker and the time-to-event process. One approach is the joint latent class model (Proust-Lima et al., 2014) that utilizes latent classes as this latent parameter. In these models n subjects are divided into different latent homogeneous subgroups/classes where each class has its own class-specific longitudinal trajectory and a class-specific risk for the event. It is assumed that the longitudinal trajectory and time-to-event are conditionally independent given these latent classes. This type of models account for heterogeneous subpopulations but permits to flexibly model the association between the longitudinal marker and event process (Proust-Lima et al., 2014). Therefore, in the following we are going to only focus on shared random effects joint models where the random effects are assumed to link the longitudinal and survival process.

Moreover, we need to differentiate between joint models that only include one longitudinal marker and multivariate joint models that contain multiple longitudinal markers. Due to the complexity when modeling multiple longitudinal markers, we are only going to focus on joint models that consider one longitudinal marker for the rest of this work.

In the following, a standard joint model setup together with a frequentist estimation approach is presented, closely following Rizopoulos (2012). In Section 2.3.2 we are going to give an overview on the Bayesian joint models approach, referring to Rizopoulos (2016).

2.3.1 Model formulation

As in Sections 2.1 and 2.2, for subject i, $i = 1, 2, ..., n$, let T_i^* and C_i denote the true event time and censoring time, respectively; let $T_i = \min(T_i^*, C_i)$ be the observed event time and $\delta_i = I(T_i^* \leq C_i)$ being the event indicator. For the internal time-varying covariate let $y_i(t)$ denote its value at time point t for subject i. Values for y_i are only observed intermittently at time points t_{ij} with $j = 1, ..., n_i$. Hence, the actual observed longitudinal data consist of $y_{ij} = \{y_i(t_{ij}), j = 1, ..., n_i\}$. Following Rizopoulos (2012) the basic joint model consists of a survival submodel which is of the form

$$h_i(t) = h_0(t) \exp\left\{\boldsymbol{w}_i^\top \boldsymbol{\gamma} + \alpha m_i(t)\right\}, \quad t > 0, \tag{2.20}$$

with baseline survival covariates $\boldsymbol{w}_i$ and the 'true' longitudinal marker $m_i(t)$ where α models the association between the longitudinal and survival process. This true marker value is obtained from y_{ij} by using a linear mixed model as in (2.1)

$$\begin{aligned} y_{ij} &= m_i(t_{ij}) + \varepsilon_{ij} \\ &= \boldsymbol{x}_i(t_{ij})^\top \boldsymbol{\beta} + \boldsymbol{z}_i(t_{ij})^\top \boldsymbol{b}_i + \varepsilon_{ij}, \end{aligned} \tag{2.21}$$

with $\boldsymbol{x}_i(t_{ij})$ denoting the vector of covariates for the fixed effects and $\boldsymbol{z}_i(t_{ij})$ the vector of covariates for the random effects for subject i at time point t_{ij} and $\varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$. The marker enters the model as a continuously defined covariate in time without measurement error which alleviates two of the previously mentioned issues. In the case of joint models a specification of the baseline hazard $h_0(.)$ is required since otherwise computation of standard errors is complicated. For this we commonly use e.g. a parametric distribution, piecewise-constant functions or a spline-based approach (for more details see Chapter 4.3 in Rizopoulos (2012)).

In this Section we base the estimation of the joint model's parameters on a maximum likelihood approach that maximizes the joint likelihood of the survival and longitudinal process $\{\boldsymbol{T}, \boldsymbol{\delta}, \boldsymbol{y}\}$ with $\boldsymbol{T} = [T_1, ..., T_n]^\top$ and $\boldsymbol{\delta} = [\delta_1, ..., \delta_n]^\top$. To derive the joint likelihood we assume conditional independence of both, the two submodels and the repeated observations over time, given the random effects, that is

$$p(T_i, \delta_i, \boldsymbol{y}_i \mid \boldsymbol{b}_i; \boldsymbol{\vartheta}) = p(T_i, \delta_i \mid \boldsymbol{b}_i; \boldsymbol{\vartheta})\, p(\boldsymbol{y}_i \mid \boldsymbol{b}_i; \boldsymbol{\vartheta}), \quad \text{and} \tag{2.22}$$

$$p(\boldsymbol{y}_i \mid \boldsymbol{b}_i; \boldsymbol{\vartheta}) = \prod_j p\{y_i(t_{ij}) \mid \boldsymbol{b}_i; \boldsymbol{\vartheta}\}, \tag{2.23}$$

where $\boldsymbol{\vartheta} = [\boldsymbol{\vartheta}_t^\top, \boldsymbol{\vartheta}_y^\top, \boldsymbol{\vartheta}_b^\top]^\top$ denotes the full parameter vector of the joint model with the parameters of the survival model $\boldsymbol{\vartheta}_t$, the longitudinal model $\boldsymbol{\vartheta}_y$, and the parameters $\boldsymbol{\vartheta}_b$ for the covariance matrix of the random effects. Using this independence assumption we can formally derive the likelihood in a shared parameter approach as

$$\begin{aligned}
L(\boldsymbol{\vartheta} \mid \boldsymbol{T}, \boldsymbol{\delta}, \boldsymbol{y}) &= \prod_{i=1}^{n} p(T_i, \delta_i, \boldsymbol{y}_i \mid \boldsymbol{\vartheta}) \\
&= \prod_{i=1}^{n} \int p(T_i, \delta_i, \boldsymbol{y}_i, \boldsymbol{b}_i \mid \boldsymbol{\vartheta}) d\boldsymbol{b}_i \\
&= \prod_{i=1}^{n} \int p(T_i, \delta_i, \boldsymbol{y}_i \mid \boldsymbol{b}_i; \boldsymbol{\vartheta})\, p(\boldsymbol{b_i}; \boldsymbol{\vartheta}_b) d\boldsymbol{b}_i \\
&= \prod_{i=1}^{n} \int p(T_i, \delta_i \mid \boldsymbol{b}_i; \boldsymbol{\vartheta}_t)\, p(\boldsymbol{y}_i \mid \boldsymbol{b}_i; \boldsymbol{\vartheta}_y)\, p(\boldsymbol{b}_i, \boldsymbol{\vartheta}_b) d\boldsymbol{b}_i \\
&= \prod_{i=1}^{n} \int p(T_i, \delta_i \mid \boldsymbol{b}_i; \boldsymbol{\vartheta}_t) \left[\prod_{j=1}^{n_i} p(y_{ij} \mid \boldsymbol{b}_i; \boldsymbol{\vartheta}_y) \right] p(\boldsymbol{b}_i; \boldsymbol{\vartheta}_b) d\boldsymbol{b}_i,
\end{aligned} \tag{2.24}$$

where the fourth equality follows from assumption (2.22) and the fifth by using assumption (2.23). The contribution of subject i to the likelihood for the survival submodel is

$$p(T_i, \delta_i \mid \boldsymbol{b}_i; \boldsymbol{\vartheta}_t) = h_i(T_i \mid \boldsymbol{b}_i; \boldsymbol{\vartheta}_t)^{\delta_i} \exp\left[- \int_0^{T_i} h_i(s \mid \boldsymbol{b}_i; \boldsymbol{\vartheta}_t) ds \right], \tag{2.25}$$

and the contribution for the longitudinal part

$$p(\boldsymbol{y}_i \mid \boldsymbol{b}_i; \boldsymbol{\vartheta}_y) = \prod_{j=1}^{n_i} \left(2\pi\sigma^2\right)^{-\frac{1}{2}} \exp\left(- \frac{\left(y_{ij} - m_i(t_{ij})\right)^2}{2\sigma^2} \right), \tag{2.26}$$

and a multivariate normal density for $p(\boldsymbol{b}_i; \boldsymbol{\vartheta}_b)$.

In a frequentist approach estimation is based on maximizing the joint likelihood (2.24). This maximization can be achieved by using standard algorithms. In the literature of joint models the Expectation Maximization (EM) algorithm, where the random effects are treated as 'missing data', is traditionally preferred (Wulfsohn and Tsiatis, 1997; Henderson et al., 2000). A drawback of the EM algorithm is its slow convergence especially near the maximum. Therefore, Rizopoulos (2012) suggests the use of a combination of EM and Newton-Raphson algorithm to achieve faster convergence. For quick introductions to the EM and Newton-Raphson algorithm see B.3 and B.2 in the appendix.

Moreover, the maximization of the joint likelihood is complicated by the necessity of numerical integration for both integrals, the one over time in the survival likelihood (2.25) and the one with respect to the random effects in (2.24). The integral over time is always one-dimensional. However, the integration over the random effects causes difficulties since highly flexible subject-specific structures increase the dimension of the random effects which makes the estimation of the parameters unstable and may even be infeasible in a frequentist setup. Therefore, the integrals are approximated using a Gaussian quadrature. For a brief introduction to Gaussian quadrature see B.1.

In order to estimate a shared random effects joint model that contains one longitudinal marker in a frequentist setting, different software implementations have already been developed, such as the R packages `JM` (Rizopoulos, 2010), `joineR` (Philipson et al., 2020), the stata module `stjm` (Crowther, 2013), and the SAS macro `JMFit` (Zhang et al., 2016). For a more detailed overview on the existing software see Papageorgiou et al. (2019). Note that the packages highly differ in their flexibility allowing to model the baseline hazard, the longitudinal trajectory, as well as the association between the longitudinal and time-to-event outcome. Moreover, they differently parameterize the association between the longitudinal marker and the time-to-event process, i.e. some only include the random effects $\boldsymbol{b}_i$ in the model for the survival outcome whereas other packages include the complete true longitudinal measure m_i.

2.3.2 Bayesian analysis in joint models

The joint model introduced so far is based on a frequentist approach. However, it often can be beneficial to apply a Bayesian approach, since a Bayesian joint model approach allows straightforward model assessment, asymptotic approximations for inference are not necessary and prior beliefs on the parameters can be incorporated (Gould et al., 2015). This approach also allows to more flexibly model the subject-specific random effects since now the random effects are also treated as parameters and we therefore do not need to integrate them out in the likelihood. Formally this can be seen from the posterior distribution of the joint model (Rizopoulos, 2016)

$$p(\boldsymbol{\vartheta}, \boldsymbol{b} \mid \boldsymbol{y}, \boldsymbol{T}, \boldsymbol{\delta}) \propto \prod_{i=1}^{n} p(T_i, \delta_i \mid \boldsymbol{b}_i, \boldsymbol{\vartheta}_t) \left[\prod_{j=1}^{n_i} p(y_{ij} \mid \boldsymbol{b}_i; \boldsymbol{\vartheta}_y) \right] p(\boldsymbol{b}_i; \boldsymbol{\vartheta}_b) p(\boldsymbol{\vartheta}). \tag{2.27}$$

Now also the parameters $\boldsymbol{\vartheta}_y$ and $\boldsymbol{\vartheta}_t$ are assumed to be random variables with prior distribution $p(\boldsymbol{\vartheta})$. Point estimates for the parameters of the joint model can be obtained by posterior mode or posterior mean estimation which are typically calculated by either Newton-Raphson algorithm or Markov Chain Monte Carlo (MCMC) sampling, respectively. The most widely used sampler in Bayesian joint models is the Gibbs sampling approach, see for instance Faucett and Thomas (1996); R. Brown and G. Ibrahim (2003), where the full parameter vector $\boldsymbol{\vartheta}$ is divided into P possibly multivariate parameter blocks $\boldsymbol{\vartheta}_p$. For each of these blocks the full conditionals $f(\boldsymbol{\vartheta}_p \mid \boldsymbol{y}, \boldsymbol{\vartheta}_{-p})$, with

$\boldsymbol{\vartheta}_{-p}$ denoting all parameter blocks but the p -th, are known. In every iteration step $l = 1, ..., L$, the sampler loops threw all P blocks and draws a sample $\vartheta_p^{(l)}$ based on all recent samples of ϑ_{-p}.

So far only two R packages, `JMBayes` (Rizopoulos, 2016) and `bamlss` (Umlauf et al., 2018), have been developed that are able to fit a joint model under a Bayesian approach. The main difference between those two packages is the flexibility allowing to model the subject-specific longitudinal trajectories and non-linear effects.

3 Flexible Bayesian additive joint models

In the following, a flexible framework for estimating additive joint models is presented where we are especially going to focus on two extensions of the former introduced standard shared parameter joint model: a more flexible specification of the longitudinal trajectory and a time-varying association of the longitudinal and time-to-event process. Therefore, we are first going to introduce the concept of B-splines, and especially its penalized version P-splines, which allow us to more flexibly model the effect of the covariates on the response. Then we will present the general setup and estimation of the flexible Bayesian additive joint models where we closely follow Köhler et al. (2017) who developed and implemented this type of models in the R package `bamlss`.

3.1 Penalized B-splines

In order to achieve a more flexible modeling of the relationship between the covariates and the response we want to relax the linearity assumption. Assuming linearity we have the following effect from one covariate on the response

$$E(\boldsymbol{y} \mid \boldsymbol{x}) = \boldsymbol{x}\beta, \tag{3.1}$$

with the observed response $\boldsymbol{y} = [y_1, ..., y_n]^\top$, the observed covariate $\boldsymbol{x} = [x_1, ..., x_n]^\top$ as well as the unobserved coefficient β. A first step to gain more flexibility could be the use of polynomials. However, a polynomial of small order implies a specific form of f which thus lacks in flexibility. A higher polynomial allows more flexibility though, but often leads to wiggly functions and variable estimates. To overcome these problems we use a nonparametric regression where we assume that the dependence of $\boldsymbol{y}$ on $\boldsymbol{x}$ is given by

$$E(\boldsymbol{y} \mid \boldsymbol{x}) = f(\boldsymbol{x}), \tag{3.2}$$

where $f(.)$ is an unspecified smooth function. There exist several approaches on how to define this function. In this work we will only focus on a spline approach, specifically the penalized B-splines representation.

B-splines (De Boor, 1978) are piecewise polynomial functions of degree l that are smoothly joined at a sequence of m knots $\kappa_1, ..., \kappa_m$. Each basis function has a support that covers $l+2$ knots. Using B-splines, we can define the smooth function in (3.2) as a linear combination (Fahrmeir and Tutz, 2013)

$$f(x_i) = \sum_{d=1}^{D} \beta_d B_d(x_i), \tag{3.3}$$

where $B_d(x_i)$ denotes the dth spline basis function evaluated at x_i and β_d is the corresponding coefficient. A B-spline basis of degree l yields $D = l + m - 1$ basis functions which can be recursively defined as

$$B_d^l(z) = \frac{z - \kappa_{d-l}}{\kappa_d - \kappa_{d-l}} B_{d-1}^{l-1} + \frac{\kappa_{d+l} - z}{\kappa_{d+1} - \kappa_{d+1-l}} B_d^{l-1}(z). \tag{3.4}$$

B-splines can be incorporated in a linear model by setting up the following design matrix

$$\boldsymbol{B} = \begin{pmatrix} B_1(x_1) & \cdots & B_D(x_1) \\ \vdots & & \vdots \\ B_1(x_n) & \cdots & B_D(x_n) \end{pmatrix}, \tag{3.5}$$

where the corresponding parameter vector $\boldsymbol{\beta} = [\beta_1, ..., \beta_D]^\top$ can be estimated using the least squares criterion. A common choice of basis functions are cubic splines. The more difficult task is deciding for a number and position of knots since they strongly determine the degree of smoothness. The knots may be placed equidistant or by quantiles (Fahrmeir and Tutz, 2013).

For a large number of knots, the resulting function could become too wiggly, resulting in overfitting. Whereas for a smaller number the function may be too flat. To overcome this trade-off between over- and underfitting and in order to avoid choosing an explicit number of knots, Eilers and Marx (1996) suggest the use of penalized B-splines, so-called P-splines, which penalize abrupt jumps between neighboring spline coefficients by introducing a difference penalty. Therefore, typically a large number of equidistant knots is selected. Instead of fitting by least squares we now minimize the penalized least squares criterion

$$PLS(\lambda) = \sum_{i=1}^{n} \left(y_i - \sum_{d=1}^{D} \beta_d B_d(x_i) \right)^2 + \lambda \sum_{d=r+1}^{D} (\Delta_r \beta_d)^2, \tag{3.6}$$

where the difference penalties $\Delta_r \beta_d = \Delta_r \beta_d - \Delta_{r-1} \beta_{d-1}$ are recursively defined (Fahrmeir and Tutz, 2013). A popular penalty are second order differences since they correspond to a penalty on the second derivative of the spline which penalizes too strong curvature of the function. For a matrix representation of the difference penalty we define the difference matrices $\boldsymbol{D}_r$

$$\boldsymbol{D}_1 = \begin{pmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{pmatrix} \qquad \boldsymbol{D}_2 = \begin{pmatrix} 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -2 & 1 \end{pmatrix} \tag{3.7}$$

where $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ are difference matrices of first and second order, respectively. The penalty term from the above PLS criterion can then be rewritten as

$$\lambda \sum_{d=r+1}^{D} (\Delta_r \beta_d)^2 = \lambda \boldsymbol{\beta}^\top \boldsymbol{D}_r^\top \boldsymbol{D}_r \boldsymbol{\beta} = \lambda \boldsymbol{\beta}^\top \boldsymbol{K}_r \boldsymbol{\beta}. \tag{3.8}$$

The smoothing parameter λ controls the amount of smoothness, which can for instance be estimated by minimizing the AIC criterion or cross-validation (Fahrmeir and Tutz, 2013). For $\lambda \to 0$, meaning there is barely any penalization, we get a function close to standard B-splines. For $\lambda \to \infty$, which means giving a large weight to the penalty of order r, the resulting function will approach a polynomial of degree $r - 1$ (Eilers and Marx, 1996). Second order penalties would then for example lead to a straight line.

It is also possible to estimate P-splines in a Bayesian approach as Bayesian P-splines (Lang and Brezger, 2004) where the spline coefficients $\boldsymbol{\beta}$ are now assumed to be random variables and smoothness is caused by an appropriate prior distribution. More accurately, the difference penalties

are induced by their stochastic analogue: a random walk. A first order difference penalty for instance corresponds to a first order random walk and second differences to a second order random walk (Lang and Brezger, 2004)

$$\beta_d = \beta_{d-1} + u_d \quad \text{or} \quad \beta_d = 2\beta_{d-1} - \beta_{d-2} + u_d, \quad d = 2, ..., D \tag{3.9}$$

with the Gaussian error $u_d \sim \mathcal{N}(0, \tau^2)$ and $\beta_1 \sim const$, or $\beta_1 \sim const$ and $\beta_2 \sim const$ as the priors for the starting value in a first order or second order random walk, respectively. For a first order random walk we can derive the conditional distribution of β_d (Fahrmeir et al., 2009) as

$$\beta_d \mid \beta_{d-1}, ..., \beta_1 \sim \mathcal{N}(\beta_{d-1}, \tau^2) \tag{3.10}$$

which implies that the conditional expectation of β_d only depends on the previous coefficient β_{d-1}. Moreover, the larger the variance parameter τ^2 the larger we allow β_d to deviate from β_{d-1}. Thus, τ^2 controls the amount of smoothness and corresponds to the inverse smoothing parameter λ in a frequentist approach. Using the conditional distribution (3.10) and the fact that we have a constant prior for the starting value β_1, we can derive a smoothness prior for a first order random walk (Fahrmeir et al., 2009), that is,

$$\begin{aligned}
p(\boldsymbol{\beta} \mid \tau^2) &= \prod_{d=1}^{D} p(\beta_d \mid \beta_{d-1}, ..., \beta_1) \\
&= p(\beta_1) \prod_{d=2}^{D} p(\beta_d \mid \beta_{d-1}) \\
&\propto \prod_{d=2}^{D} \frac{1}{\sqrt{2\pi\tau^2}} \exp\left(-\frac{1}{2\tau^2}(\beta_d - \beta_{d-1})^2\right) \\
&\propto \exp\left(-\frac{1}{2\tau^2} \sum_{d=1}^{D} (\beta_d - \beta_{d-1})^2\right) \\
&\propto \exp\left(-\frac{1}{2\tau^2} \boldsymbol{\beta}^\top \boldsymbol{K}_1 \boldsymbol{\beta}\right),
\end{aligned} \tag{3.11}$$

with the previously defined precision matrix $\boldsymbol{K}_1 = \boldsymbol{D}_1^\top \boldsymbol{D}_1$. Note that $rank(\boldsymbol{K}_1) = D - 1$ and thus the prior is partially improper. The prior for a second order difference penalty can be derived analogously and equals the prior in (3.11), except for the precision matrix $\boldsymbol{K}$, up to a multiplicative constant.

3.2 General setup

In line with Section 2, let T_i denote the probably right-censored observed event time and δ_i be the event indicator that equals 1 if subject i experiences the event. The hazard for an event at some time point t is modeled as

$$h_i(t) = \exp\{\eta_i(t)\} = \exp\{\eta_{\lambda i}(t) + \eta_{\gamma i} + \eta_{\alpha i}(t) \cdot \eta_{\mu i}(t)\}, \tag{3.12}$$

with the full predictor η including a predictor η_λ for all time-varying survival covariates and the log-baseline hazard, a predictor η_γ for all baseline survival covariates, and the longitudinal measure

η_μ that is related to the hazard via the possibly time-varying predictor η_α (Köhler et al., 2017). The longitudinal predictor $\eta_{\mu i}$ is modeled using the longitudinal responses y_{ij} as

$$y_{ij} = \eta_{\mu i}(t_{ij}) + \varepsilon_{ij}, \tag{3.13}$$

where we can also model the independent residuals $\varepsilon_{ij} \sim \mathcal{N}(0, \exp[\eta_{\sigma i}(t_{ij})]^2)$ via the predictor $\eta_{\sigma i}$. The predictor $\eta_{\mu i}(t_{ij})$ represents the 'true' longitudinal value at time point t_{ij} without the potential measurement error. This 'true' value links the two models (3.12) and (3.13) and further it is a continuous time-varying covariate in the definition of the hazard.

In order to gain more flexibility, each predictor η_{ki} with $k \in \{\lambda, \gamma, \alpha, \mu, \sigma\}$ can be modeled as a sum of M_k potentially nonparametric functions (see previous Section 3.1). Formally, we have the structured additive predictors

$$\eta_{ki} = \sum_{m=1}^{M_k} f_{km}(\boldsymbol{x}_{ki}), \tag{3.14}$$

with the covariates $\boldsymbol{x}_{ki}$ where each function usually depends on either one or two covariates. Moreover, for time-dependent predictors the functions f can also depend on time including a possibly time-varying covariate vector $\boldsymbol{x}_{ki}(t)$. Let the vector $\boldsymbol{\eta}_k = [\eta_{k1}, \eta_{k2}, ..., \eta_{kn}]^\top$ contain the predictors for all subjects. Then, the vectors in the survival submodel (3.12) are of length n where $\boldsymbol{\eta}_k(t)$ holds the evaluations for all n subjects at time point t. In the longitudinal submodel (3.13), the vector $\boldsymbol{\eta}_k(\boldsymbol{t})$ is of length $N = \sum_i n_i$, evaluated at $\boldsymbol{t} = [\boldsymbol{t}_1^\top, \boldsymbol{t}_2^\top, ..., \boldsymbol{t}_n^\top]^\top$ which is a vector that contains the stacked observation time points $\boldsymbol{t}_i = [t_{i1}, t_{i2}, ..., t_{in_i}]^\top$ for all corresponding subjects.

Using these structured additive predictors allow us to model a variety of effects, including linear, time-varying, smooth, spatial or random effects. To model these effects we use suitable basis functions, that could be for example a spline basis, and its resulting design matrix $\boldsymbol{X}_{km}$ in connection with a penalty matrix $\boldsymbol{P}_{km}$ for term m of predictor k. This yields the generic setup

$$\boldsymbol{f}_{km} = \boldsymbol{X}_{km}\boldsymbol{\beta}_{km} \qquad \text{and} \qquad \boldsymbol{P}_{km} = \frac{1}{\tau_{km}^2}\boldsymbol{\beta}_{km}^\top \boldsymbol{K}_{km}\boldsymbol{\beta}_{km}. \tag{3.15}$$

Here, $\boldsymbol{f}_{km}$ denotes the piled function evaluations over individuals, $\boldsymbol{X}_{km}$ are the design matrices of dimension $n \times p_{km}$ or $N \times p_{km}$ for the survival and longitudinal submodel, respectively; and $\boldsymbol{\beta}_{km}$ is a coefficient vector of length p_{km}. Further, $\boldsymbol{K}_{km}$ denotes a precision matrix, that shrinks the corresponding vector $\boldsymbol{\beta}_{km}$ towards zero or penalizes sudden jumps among neighboring parameters. The variance parameters τ_{km}^2 control the amount of smoothness for the respective term. For a Bayesian estimation approach, penalization is imposed by defining appropriate prior distributions for the parameters. Hence, the penalty $\boldsymbol{P}_{km}$ needs to be incorporated into the prior which for example could be $\boldsymbol{\beta}_{km} \sim \mathcal{N}(\boldsymbol{0}, [\frac{1}{\tau_{km}^2}\boldsymbol{K}_{km}]^-)$ with $\boldsymbol{A}^-$ being the generalized inverse of $\boldsymbol{A}$. The unknown variance parameters τ_{km} are either estimated in a fully Bayesian approach by specifying appropriate hyperpriors such as inverse Gamma priors, or for empirical Bayes inference the variance parameters are considered unknown constants which are estimated from the data via ML.

To illustrate this setup we will show two examples: First, we can use the former introduced P-splines to model smooth functions in time where we use a B-splines basis, $f_{km}(t) = \sum_{d=1}^{D} \beta_d \boldsymbol{B}_d(t) =: \tilde{\boldsymbol{x}}_{km}^\top(t)\boldsymbol{\beta}_{km}$, where $\tilde{\boldsymbol{x}}_{km}$ denotes the resulting design vector for covariates $\boldsymbol{x}_{km}$, together with a difference precision matrix $\boldsymbol{K}_{km} = \boldsymbol{D}_r^\top \boldsymbol{D}_r$ with the r-th difference matrix $\boldsymbol{D}_r$ of appropriate dimension. This approach allows us to avoid deciding on an explicit number of knots and takes under consideration the trade-off between over- and underfitting. Second, we can model a random intercept in the longitudinal predictor for each subject by using a basis function $\boldsymbol{X}_{\mu m}$ of dimension $N \times n$ where the ith column indicates which longitudinal observations belong to subject i. The corresponding

precision matrix is an n dimensional identity matrix, $K_{\mu m} = I_n$. Thus, using the above prior we get $\beta_{\mu mi} \sim \mathcal{N}(0, \tau^2_{km})$ independently.

3.3 Important extensions

Longitudinal trajectories over time are often heterogeneous between subjects which calls for a highly flexible and subject-specific modeling (compare Figure 1). Thus, the longitudinal predictor η_μ can be modeled as (Köhler et al., 2017)

$$\eta_{\mu i}(t) = f_{\mu 1}(t) + f_{\mu 2}(i) + f_{\mu 3}(t, i) + \sum_{m=4}^{M_\mu} f_{\mu m}(\boldsymbol{x}_{\mu m}), \tag{3.16}$$

where $f_{\mu 1}(t)$ is a smooth effect of time, $f_{\mu 2}(i)$ models a subject-specific random intercept, and $f_{\mu 3}(t, i)$ denotes subject-specific deviations from the overall time effect using functional random intercepts (Scheipl et al., 2015). Besides, additional smooth, linear or parametric effects, including a global intercept, may be specified within the term $\sum_{m=4}^{M_\mu} f_{\mu m}(\boldsymbol{x}_{\mu m})$.

The corresponding basis matrix $\boldsymbol{X}_{\mu 3}$ for the functional random intercepts is constructed using a row tensor product, denoted by $\odot$, of the marginal basis matrix for the random subject-specific intercepts $\boldsymbol{X}_{\mu 3s}$ and the marginal basis for the smooth effect of time $\boldsymbol{X}_{\mu 3t}$. In the following, we are going to mark all parameters or matrices related to the random intercept with a subscript s and those related to the smooth effect of time via subscript t. Using the row tensor product we get

$$\boldsymbol{X}_{\mu 3} = \boldsymbol{X}_{\mu 3s} \odot \boldsymbol{X}_{\mu 3t} = (\boldsymbol{X}_{\mu 3s} \otimes \boldsymbol{1}_D^\top) \cdot (\boldsymbol{1}_n^\top \otimes \boldsymbol{X}_{\mu 3t}), \tag{3.17}$$

with the $N \times n$ indicator matrix $\boldsymbol{X}_{\mu 3s}$ and the $N \times D$ matrix $\boldsymbol{X}_{\mu 3t}$ containing the spline basis functions of the smooth time effect evaluated at $\boldsymbol{t}$. $\cdot$ denotes element wise multiplication and $\otimes$ is the Kronecker product. The resulting basis for the functional random intercepts $\boldsymbol{X}_{\mu 3}$ is an $N \times nD$ matrix. This leads to the stacked function evaluations $\boldsymbol{f}_{\mu 3} = \boldsymbol{X}_{\mu 3}\boldsymbol{\beta}_{\mu 3}$ with the coefficient vector $\boldsymbol{\beta}_{\mu 3}$ of length nD. The corresponding penalty $\boldsymbol{P}_{\mu 3}$ is constructed using both marginal precision matrices:

$$\boldsymbol{P}_{\mu 3} = \boldsymbol{\beta}_{\mu 3}^\top \left(\frac{1}{\tau^2_{\mu 3s}} \boldsymbol{K}_{\mu 3s} \otimes \boldsymbol{I}_t + \frac{1}{\tau^2_{\mu 3t}} \boldsymbol{I}_s \otimes \boldsymbol{K}_{\mu 3t} \right) \boldsymbol{\beta}_{\mu 3} = \boldsymbol{\beta}_{\mu 3}^\top \left(\frac{1}{\tau^2_{\mu 3s}} \tilde{\boldsymbol{K}}_{\mu 3s} + \frac{1}{\tau^2_{\mu 3t}} \tilde{\boldsymbol{K}}_{\mu 3t} \right) \boldsymbol{\beta}_{\mu 3}, \tag{3.18}$$

where $\boldsymbol{K}_{\mu 3s} = \boldsymbol{I}_n$ is the precision matrix for the subject-specific random intercepts and $\boldsymbol{K}_{\mu 3t}$ is an appropriate precision matrix for the marginal smooth effect of time that can be for instance a difference penalty. $\boldsymbol{I}_t$ and $\boldsymbol{I}_s$ are identity matrices of dimension D and n, respectively. The Kronecker product then leads to the inflated matrices $\tilde{\boldsymbol{K}}_{\mu 3s}$ and $\tilde{\boldsymbol{K}}_{\mu 3t}$ which are of size $nD \times nD$ yielding a penalization for every subject as well as a smoothness penalization throughout time. $\tau^2_{\mu 3s}$ and $\tau^2_{\mu 3t}$ are the variance parameters for the random intercepts and the smooth effect of time, respectively, which control the degree of penalization. Note that $\tau_{\mu 3s}$ controls the amount of penalization for the subject-specific random effects structure whereas $\tau_{\mu 3t}$ controls the time effect penalization. Hence, the amount of penalization can differ in both directions: across subjects and time.

When specifying a global intercept or additional subject-specific random intercepts, we need to set the following constraints to guarantee indentifiability of the parameters: $\int f_{\mu 1}(t)dt = 0$ and $\int f_{\mu 3}(t, i)dt = 0$ for each subject i. The global intercept for the survival model is typically contained in η_μ. For more details on the implementation of the constraints see Köhler et al. (2017).

A second extension to the basic shared parameter joint model is that we are now able to model the association between the longitudinal measures and the time-to-event process as an additive predictor η_α. Consequently, the association can be modeled as a function that depends on time leading to a time-varying association, or depend on other covariates. This time-varying association between the biomarker and event process is beneficial since, especially when analysing the course of a disease, the biomarker is in interaction with an ongoing immune process and therefore it is reasonable to assume a time-varying effect from the biomarker on the event. This way of modeling the association also permits to detect time slots in which we have a significant effect of the biomarker on the event time, or a potential change in the direction of the effect over time (Köhler et al., 2017).

A further extension of the presented model is that we can additionally model the error variance ε of the longitudinal submodel using the predictor η_σ. Hence, the error variance can depend on time or other covariates.

3.4 Estimation

The estimation of the joint model is based on a Bayesian approach using Newton-Raphson algorithm and MCMC sampling. This is the case since the possibly highly flexible random effects structure increase the dimension of its prior distribution. A frequentist approach requires to integrate over this potentially high dimensional distribution which may lead to infeasible estimates. Thus, to avoid those problems a Bayesian approach is used.

As for the standard joint model in Section 2.3, the joint likelihood of the longitudinal and time-to-event process can be derived as the product of the likelihoods of both submodels where we are using the assumption of conditional independence of the time-to-event outcome $[T_i, \delta_i]$ and longitudinal outcome $\boldsymbol{y}_i$. The log-likelihood for the time-to-event submodel is

$$l^{\text{event}}[\boldsymbol{\eta}_\lambda(\boldsymbol{T}), \boldsymbol{\eta}_\gamma, \boldsymbol{\eta}_\alpha(\boldsymbol{T})\boldsymbol{\eta}_\mu(\boldsymbol{T}) \mid \boldsymbol{T}, \boldsymbol{\delta}] = \boldsymbol{\delta}^\top \boldsymbol{\eta}(\boldsymbol{T}) - \mathbf{1}_n^\top \boldsymbol{\Lambda}(\boldsymbol{T}), \tag{3.19}$$

with $\boldsymbol{T} = [T_1, ..., T_n]^\top$ and $\boldsymbol{\delta} = [\delta_1, ..., \delta_n]^\top$. Furthermore, $\boldsymbol{\Lambda}(\boldsymbol{T}) = [\Lambda_1(T_1), ..., \Lambda_n(T_n)]^\top$ denotes the vector of the cumulative hazard rates $\Lambda_i(T_i) = \exp(\eta_{\gamma i}) \int_0^{T_i} \exp[\eta_{\lambda i}(u) + \eta_{\alpha i}(u) \cdot \eta_{\mu i}(u)]du$ for all subjects and $\boldsymbol{\eta}(\boldsymbol{T})$ is the vector of full predictors evaluated at the subject-specific observed event times. The log-likelihood for the longitudinal submodel is

$$l^{\text{long}}[\boldsymbol{\eta}_\mu(\boldsymbol{t}), \boldsymbol{\eta}_\sigma(\boldsymbol{t}) \mid \boldsymbol{y}] = -\frac{N}{2}\log(2\pi) - \mathbf{1}_N^\top \boldsymbol{\eta}_\sigma(\boldsymbol{t}) - \frac{1}{2}(\boldsymbol{y} - \boldsymbol{\eta}_\mu(\boldsymbol{t}))^\top \boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{\eta}_\mu(\boldsymbol{t})), \tag{3.20}$$

where $\boldsymbol{y} = [\boldsymbol{y}_1^\top, ..., \boldsymbol{y}_n^\top]^\top$ is the longitudinal response and $\boldsymbol{\eta}_\mu(\boldsymbol{t})$ and $\boldsymbol{\eta}_\sigma(\boldsymbol{t})$ are the corresponding predictor values of length N. The variance term $\boldsymbol{R} = \text{blockdiag}(\boldsymbol{R}_1, ..., \boldsymbol{R}_n)$ with $\boldsymbol{R}_i$ reflecting the error structure which we assume to be $\boldsymbol{R}_i = \text{diag}(\exp[\eta_{\sigma i}(t_{i1})]^2, ..., \exp[\eta_{\sigma i}(t_{in_i})]^2)$.

The posterior can be derived by specifying appropriate prior distributions for the parameters. Due to the flexible framework and the possibility of specifying a variety of terms (parametric, linear, smooth, etc.) the priors differ for the different specifications. Thus, vague normal priors are assigned to linear or parametric terms and multivariate normal priors are used for random or smooth terms. A more detailed overview of suitable priors for the different terms can be found in Köhler et al. (2017). The resulting posterior of the joint model is then

$$p(\boldsymbol{\vartheta} \mid \boldsymbol{T}, \boldsymbol{\delta}, \boldsymbol{y}) \propto L^{\text{event}}[\boldsymbol{\eta}_\lambda(\boldsymbol{T}), \boldsymbol{\eta}_\gamma, \boldsymbol{\eta}_\alpha(\boldsymbol{T}), \boldsymbol{\eta}_\mu(\boldsymbol{T}) \mid \boldsymbol{T}, \boldsymbol{\delta}] \cdot L^{\text{long}}[\boldsymbol{\eta}_\mu(\boldsymbol{t}), \boldsymbol{\eta}_\sigma(\boldsymbol{t}) \mid \boldsymbol{y}]$$
$$\prod_{k \in \{\lambda, \gamma, \alpha, \mu, \sigma\}} \prod_{m=1}^{M_k} [p(\boldsymbol{\beta}_{km} \mid \boldsymbol{\tau}_{km}^2) p(\boldsymbol{\tau}_{km}^2)], \tag{3.21}$$

where L^{event} and L^{long} are the likelihoods for the time-to-event and longitudinal submodels, respectively; $\boldsymbol{\vartheta}$ is the vector of all parameters in the model and $p(\boldsymbol{\beta}_{km} \mid \boldsymbol{\tau}^2_{km})$ and $p(\boldsymbol{\tau}^2_{km})$ are the corresponding priors for the regression coefficients and the variance parameters, respectively. For anisotropic smooths, as for example for the functional random intercepts, we can have multiple variance parameters $\boldsymbol{\tau}^2_{km} = (\tau^2_{kms}, \tau^2_{kmt})$ involved.

Point estimates for $\boldsymbol{\vartheta}$ can be obtained by posterior mode or posterior mean estimation. Due to the possibility of modeling highly flexible subject-specific structures the posterior mean estimation can be relatively time consuming. Therefore, posterior mode estimates can be used for a first evaluation of the model and to obtain starting values for the posterior mean sampling.

Flexible Bayesian additive joint models can be estimated in the R-package `bamlss` (Umlauf et al., 2018). Internally, they use the R-package `mgcv` (Wood, 2011) for the specification of appropriate basis matrices and corresponding penalties. In `bamlss` the posterior mode estimation is based on a Newton-Raphson procedure that updates each term m of predictor k blockwise in each iteration step l as

$$\boldsymbol{\beta}^{(l+1)}_{km} = \boldsymbol{\beta}^{(l)}_{km} - \nu^{(l)}_{km} \boldsymbol{H} \left(\boldsymbol{\beta}^{(l)}_{km}\right)^{-1} \boldsymbol{s} \left(\boldsymbol{\beta}^{(l)}_{km}\right) \tag{3.22}$$

with potentially varying steplength ν_{km} that is maximized in each step to optimize the posterior. $\boldsymbol{s}(\boldsymbol{\beta}_{km})$ denotes the corresponding score vector and $\boldsymbol{H}(\boldsymbol{\beta}_{km})$ the Hessian. For a detailed derivation of the score vector and the Hessian see the supplemantary material in Köhler et al. (2017).

For the posterior mean sampling the package `bamlss` utilizes an approximation of the Gibbs sampler where the full conditionals $\pi(\boldsymbol{\beta}_{km} \mid .)$ are based on a second order Taylor expansion of the log-posterior centered at the last state $\beta^{(m)}_{km}$. For the sampling of the variance parameters $\boldsymbol{\tau}^2_{km}$ they use a Gibbs sampler since the full conditionals follow in this case an inverse Gamma distribution, if inverse Gamma hyperpriors are used. More details on the estimation can be found in the corresponding paper Köhler et al. (2017) or in the documentation of the `bamlss` R-package.

4 Prediction in joint models

Thus far we have introduced the set up and estimation of joint models. But often the interest behind building a joint model is to provide predictions for an outcome of interest. Especially in personalized medicine, physicians need predictive tools that consider individual specific characteristics of their patient in order to individually adjust the treatment plan and improve decision making. Moreover, it is important to be able to update these predictions as soon as new information becomes available. Therefore, in this Section, we are going to focus on dynamic predictions which, based on a fitted joint model, allow to predict future outcomes for either the survival or longitudinal model. These predictions are dynamic since they can be updated as soon as new information is recorded. For both approaches we are going to introduce two types of estimator: first, a first-order estimator that allows relatively fast computation but does not provide any credibility intervals and second an estimator based on Monte Carlo sampling that therefore provides credibility intervals. We are going to show the derivation and estimation in a Bayesian approach closely following the dynamic predictions introduced in Rizopoulos (2016). For a frequentist approach we recommend Chapter 7 in Rizopoulos (2012).

In the following, we are first going to focus on predictions of survival probabilities utilizing all available information at hand and then in Section 4.2 the predictions for the longitudinal outcome are introduced. In Section 4.3 two popular measures from survival analysis that were adapted to our dynamic setting are introduced, based on which the quality of the dynamic prediction for the survival outcomes can be assessed. Then in the last Section 4.4 we explain how to use our

implemented dynamic prediction for flexible Bayesian additive joint models and give important details on its implementation.

4.1 Dynamic Predictions of survival probabilities

Based on a joint model fitted in a Bayesian approach to a random sample $\mathcal{D}_n = \{T_i, \delta_i, \boldsymbol{w}_i, \boldsymbol{y}_i; i = 1, ..., n\}$ from a target population, we are interested in deriving subject-specific survival probabilities for a new subject j from the same population, for whom a vector of longitudinal observations $\boldsymbol{y}_j = [y_{j1}, ..., y_{jn_j}]^\top$ and baseline covariates $\boldsymbol{w}_j$ are available. Having information on the endogenous longitudinal measurement, say a biomarker, until time point t, in fact, implies subject j to be event free until t. Hence, it is more relevant to focus on the conditional probability of surviving at least until some time point $u > t$, given survival until t. Formally, we are interested in estimating

$$\pi_j(u \mid t) = \Pr(T_j^* \geq u \mid T_j^* > t, \boldsymbol{y}_j(t), \boldsymbol{w}_j, \mathcal{D}_n), \quad t > 0, \tag{4.1}$$

where $\boldsymbol{y}_j(t) = [y_{j1}, ..., y_{jt}]^\top$ denotes the vector of all longitudinal observations on subject j until time point t. The dynamic nature of the prediction comes in from the fact that we can update the survival probability as soon as new information for subject j is recorded at a future time point t', $t' > t$.

In order to derive a first, first-order estimate for the subject-specific conditional survival probabilities $\pi_j(u \mid t)$ we can rewrite equation (4.1) as

$$\begin{aligned}
&\Pr(T_j^* \geq u \mid T_j^* > t, \boldsymbol{y}_j(t), \boldsymbol{w}_j; \boldsymbol{\vartheta}) \\
&= \int \Pr(T_j^* \geq u \mid T_j^* > t, \boldsymbol{y}_j(t), \boldsymbol{w}_j, \boldsymbol{b}_j; \boldsymbol{\vartheta})\, p(\boldsymbol{b}_j \mid T_j^* > t, \boldsymbol{y}_j(t); \boldsymbol{\vartheta})\, d\boldsymbol{b}_j \\
&= \int \Pr(T_j^* \geq u \mid T_j^* > t, \boldsymbol{w}_j, \boldsymbol{b}_j; \boldsymbol{\vartheta})\, p(\boldsymbol{b}_j \mid T_j^* > t, \boldsymbol{y}_j(t); \boldsymbol{\vartheta})\, d\boldsymbol{b}_j \\
&= \int \frac{S_j\{u \mid \mathcal{M}_j(u, \boldsymbol{b}_j, \boldsymbol{\vartheta}); \boldsymbol{\vartheta}\}}{S_j\{t \mid \mathcal{M}_j(t, \boldsymbol{b}_j, \boldsymbol{\vartheta}); \boldsymbol{\vartheta}\}}\, p(\boldsymbol{b}_j \mid T_j^* > t, \boldsymbol{y}_j(t); \boldsymbol{\vartheta})\, d\boldsymbol{b}_j,
\end{aligned} \tag{4.2}$$

using the conditional independence assumption (2.22) in the second equality. $\mathcal{M}_j(t, \boldsymbol{b}_j, \boldsymbol{\vartheta})$ denotes the complete history of the true unobserved longitudinal process for subject j up to time point t which is approximated by the longitudinal submodel. $S_j(.)$ is, as before, the survival function defined as

$$\begin{aligned}
S_j\big(t \mid \mathcal{M}_j(t, \boldsymbol{b}_j, \boldsymbol{\vartheta}); \boldsymbol{\vartheta}\big) &= \Pr\big(T_j^* > t \mid \mathcal{M}_j(t, \boldsymbol{b}_j, \boldsymbol{\vartheta}); \boldsymbol{\vartheta}\big) \\
&= \exp\Big\{ - \int_0^T h_j\big(s \mid \mathcal{M}_j(t, \boldsymbol{b}_j, \boldsymbol{\vartheta}); \boldsymbol{\vartheta}\big) ds \Big\}.
\end{aligned} \tag{4.3}$$

Using the parameter estimates $\hat{\boldsymbol{\vartheta}}$ from the fitted joint model that include the parameters from the survival and longitudinal submodels as well as the parameters for the covariance matrix of the random effects, we can derive a first-order estimate for $\pi_j(u \mid t)$, that is,

$$\tilde{\pi}_j(u \mid t) = \frac{S_j\{u \mid \mathcal{M}_j(u, \hat{\boldsymbol{b}}_j, \hat{\boldsymbol{\vartheta}}); \hat{\boldsymbol{\vartheta}}\}}{S_j\{t \mid \mathcal{M}_j(t, \hat{\boldsymbol{b}}_j, \hat{\boldsymbol{\vartheta}}); \hat{\boldsymbol{\vartheta}}\}} \tag{4.4}$$

where $\hat{\boldsymbol{b}}_j$ denotes the empirical Bayes estimate of $\boldsymbol{b}_j$ which is obtained by maximizing the posterior distribution of the random effects $p(\boldsymbol{b}_j \mid T_j^* > t, \boldsymbol{y}_j(t); \boldsymbol{\vartheta})$. This is typically done using a Newton-Raphson algorithm. The estimate $\hat{\boldsymbol{b}}_j$ is empirical in the sense that we do not specify any hyperpriors

for the parameters in the prior distribution of the random effects but rather estimate them using the random sample $\mathcal{D}_n$.

The benefit of $\tilde{\pi}_j(u \mid t)$ is that, besides the optimization for the empirical Bayes estimate $\hat{\boldsymbol{b}}_j$, it does not rely on any iterative procedure and thus can be computed relatively fast. However, deriving standard errors and credible intervals is quite complicated due to the fact that we have two sources of variability we need to account for: the parameter estimates $\hat{\boldsymbol{\vartheta}}$ from the fitted joint model and the empirical Bayes estimates $\hat{\boldsymbol{b}}_j$ for the random effects of the new subject j. To overcome this problem Rizopoulos (2011) and Proust-Lima and Taylor (2009) alternatively suggest the use of Monte Carlo simulation schemes. There, in order to account for the uncertainty from $\boldsymbol{\vartheta}$, the estimation of $\pi_j(u \mid t)$ is based on the corresponding posterior predictive distribution

$$\begin{aligned}
&\Pr(T_j^* \geq u \mid T_j^* > t, \boldsymbol{y}_j(t), \boldsymbol{w}_j, \mathcal{D}_n) \\
&\quad = \int \Pr(T_j^* \geq u \mid T_j^* > t, \boldsymbol{y}_j(t), \boldsymbol{w}_j; \boldsymbol{\vartheta})\, p(\boldsymbol{\vartheta} \mid \mathcal{D}_n)\, d\boldsymbol{\vartheta} \\
&\quad = \int \int \frac{S_j\{u \mid \mathcal{M}_j(u, \boldsymbol{b}_j, \boldsymbol{\vartheta}); \boldsymbol{\vartheta}\}}{S_j\{t \mid \mathcal{M}_j(t, \boldsymbol{b}_j, \boldsymbol{\vartheta}); \boldsymbol{\vartheta}\}}\, p(\boldsymbol{b}_j \mid T_j^* > t, \boldsymbol{y}_j(t); \boldsymbol{\vartheta})\, d\boldsymbol{b}_j\, p(\boldsymbol{\vartheta} \mid \mathcal{D}_n)\, d\boldsymbol{\vartheta}, \qquad (4.5)
\end{aligned}$$

where the second equality follows from plugging in equation (4.2) for the first part of the integrand. Further note that $p(\boldsymbol{\vartheta} \mid \mathcal{D}_n)$ and $p(\boldsymbol{b}_j \mid T_j^* > t, \boldsymbol{y}_j(t); \boldsymbol{\vartheta})$ are the posterior distributions for the parameters from the joint model and random effects, respectively. Using this fact, we can derive a Monte Carlo estimate for $\pi_j(u \mid t)$ that properly approximates the integrals in equation (4.5) using the following algorithm

S1: Compute empirical Bayes estimate $\hat{\boldsymbol{b}}_j$.

S2: Draw $\boldsymbol{\vartheta}^{(m)}$ from $p(\boldsymbol{\vartheta} \mid \mathcal{D}_n)$.

S3: Draw $\boldsymbol{b}_j^{(m)}$ from $p(\boldsymbol{b}_j \mid T_j^* > t, \boldsymbol{y}_j(t); \boldsymbol{\vartheta})$.

S4: Compute $\pi_j^{(m)} = S_j\big(u \mid \mathcal{M}_j(u, \boldsymbol{b}_j^{(m)}, \boldsymbol{\vartheta}^{(m)}); \boldsymbol{\vartheta}^{(m)}\big) \Big/ S_j\big(t \mid \mathcal{M}_j(t, \boldsymbol{b}_j^{(m)}, \boldsymbol{\vartheta}^{(m)}); \boldsymbol{\vartheta}^{(m)}\big)$.

S5: Repeat Steps 2-4 $m = 1, ..., M$ times.

In Step 2 we sample from the already existing MCMC sample from the estimation of the joint model. Step 3 is less straightforward since the posterior distribution of the random effects is of non-standard form. Thus, to be able to sample from the posterior we implement a Metropolis-Hastings algorithm using independent proposals from a multivariate t-distribution with four degrees of freedom, centered at the empirical Bayes estimate $\hat{\boldsymbol{b}}_j$, and with scale matrix $\hat{\text{var}}(\hat{\boldsymbol{b}}_j) = \big\{ - \partial^2 \log p\big(T_j^* > t, \boldsymbol{y}_j(t), \boldsymbol{b}; \hat{\boldsymbol{\vartheta}}\big) / \partial \boldsymbol{b}^\top \partial \boldsymbol{b} \mid_{\boldsymbol{b}=\hat{\boldsymbol{b}}_j} \big\}^{-1}$. The reason for choosing multivariate t proposals is twofold. First, Rizopoulos et al. (2008) have shown that as the number of observations n_i on one individual increase the posterior distribution of the random effects is dominated by the distribution of the linear mixed model, and thus resembles a multivariate normal distribution. And second, for smaller n_i, we assure sufficient coverage of the whole range of the parameters by using a distribution with heavier tails (Rizopoulos, 2011). Note that in the above algorithm the Metropolis-Hastings algorithm results in only M samples with neither a burn-in phase nor any thinning. Thus, it could be the case that this sequence does not converge to the true distribution although the empirical Bayes estimates should serve as good starting values. Therefore, we suggest to externally run this Metropolis-Hastings algorithm with a higher number of iterations including a potential burn-in period where an initial number of samples is discarded and further thin our sample by dismissing

all but the t-th sample. Then, we draw a sample of size M from this random sequence. For more details on the implementation of this algorithm see Section 4.4.

The realizations $\{\pi_j^{(m)}(u \mid t), m = 1, ..., M\}$ can then be used to obtain point estimates for $\pi_j(u \mid t)$, such as taking the median

$$\hat{\pi}_j(u \mid t) = \text{median}\big\{\pi_j^{(m)}(u \mid t), m = 1, ..., M\big\}, \tag{4.6}$$

or the mean

$$\hat{\pi}_j(u \mid t) = M^{-1} \sum_{l=1}^{M} \pi_j^{(m)}(u \mid t). \tag{4.7}$$

Standard errors and credibility intervals can be computed using the sample standard deviation or corresponding percentiles of the Monte Carlo sample, respectively. This estimation scheme now accounts for the variability from both estimates $\hat{\boldsymbol{\vartheta}}$ and $\hat{\boldsymbol{b}}_j$ by sampling in each iteration values for those parameters and then compute the first estimate (4.4) using the sampled values. Moreover, the estimators (4.6) and (4.7) are assumed to be more accurate than estimator (4.4) since the integrals in the definition of the posterior predictive distribution (4.5) are properly approximated (Rizopoulos, 2011).

4.2 Dynamic Predictions of longitudinal outcomes

Based on a joint model fitted to a sample $\mathcal{D}_n$, our aim is now to derive predictions of the longitudinal outcome for a new subject j. More precisely, we are interested in the expected value of the longitudinal outcome at some future time point $u > t$ given the observed responses $\boldsymbol{y}_j(t)$ until t:

$$\omega_j(u \mid t) = E\big\{y_j(u) \mid T_j^* > t, \boldsymbol{y}_j(t), \mathcal{D}_n\big\}, \quad u > t, \tag{4.8}$$

where $y_j(u)$ denotes the longitudinal outcome for subject j at time point u. Analogous to the survival prediction, we can derive a first-order estimator for $\omega_j(u \mid t)$ that allows faster computation but does not provide any measures of accuracy like credibility intervals. We obtain this estimator by rewriting equation (4.8) as

$$\begin{aligned} E\{y_j(u) \mid T_j^* > t, \boldsymbol{y}_j(t); \boldsymbol{\vartheta}\} &= \int E\{y_j(u) \mid \boldsymbol{b}_j; \boldsymbol{\vartheta}\} p(\boldsymbol{b}_j \mid T_j^* > t, \boldsymbol{y}_j(t); \boldsymbol{\vartheta}) d\boldsymbol{b}_j \\ &= \boldsymbol{x}_j^\top(u)\boldsymbol{\beta} + \boldsymbol{z}_j^\top(u)\bar{\boldsymbol{b}}_j^{(t)} \end{aligned} \tag{4.9}$$

with

$$\bar{\boldsymbol{b}}_j^{(t)} = \int \boldsymbol{b}_j p(\boldsymbol{b}_j \mid T_j^* > t, \boldsymbol{y}_j(t); \boldsymbol{\vartheta}) d\boldsymbol{b}_j \tag{4.10}$$

denoting the expected value of the random effects that is taken with respect to the corresponding posterior distribution using all observations until t. To get a feasible estimator $\boldsymbol{\beta}$ is replaced by its parameter estimates $\hat{\boldsymbol{\beta}}$ supplied by the joint model and the empirical Bayes estimates $\hat{\boldsymbol{b}}_j$ are used instead of $\boldsymbol{b}_j$, that is,

$$\tilde{\omega}_j(u \mid t) = \boldsymbol{x}_j^\top(u)\hat{\boldsymbol{\beta}} + \boldsymbol{z}_j^\top(u)\bar{\hat{\boldsymbol{b}}}_j^{(t)}. \tag{4.11}$$

In order to also obtain point-wise credibility intervals which account for the variability stemming from $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{b}}_j$, we derive a simulation scheme similar to the one previously introduced in 4.1 where the estimation of $\omega_j(u \mid t)$ is based on the corresponding posterior predictive distribution

$$\omega_j(u \mid t) = \int E\{y_j(u) \mid T_j^* > t, \boldsymbol{y}_j(t); \boldsymbol{\vartheta}\} p(\boldsymbol{\vartheta} \mid \mathcal{D}_n) d\boldsymbol{\vartheta}. \tag{4.12}$$

Here, the first part of the integrand equals equation (4.9). The resulting Monte Carlo simulation is similar to the previous one where the only difference lies in Step 4 where we compute

$$\omega_j^{(m)} = \boldsymbol{x}_j^\top(u)\boldsymbol{\beta}^{(m)} + \boldsymbol{z}_j^\top(u)\boldsymbol{b}_j^{(m)}, \tag{4.13}$$

resulting in a set of realizations $\{\omega_j^{(m)}(u \mid t), m = 1, ..., M\}$. As before, we can compute point estimates $\hat{\omega}_j(u \mid t)$ by either taking the median or mode (compare equations (4.6) and (4.7)).

The special case for the longitudinal predictions is that we are not only interested in predicting the future outcomes but also to model the biomarker's trajectory. Hence, it is also possible to apply the above algorithm for $u < t$ which holds since we, in fact, do not make use of the assumption $u > t$ in the derivation of this simulation scheme. We split the credibility intervals into two distinct parts where we will call the precision intervals during the observation time credibility intervals and the intervals for the prediction part are denoted as prediction intervals. To derive point-wise credibility intervals we will as before take the corresponding quantiles from the set of realizations. For the prediction intervals we will take the corresponding quantiles from $\{\bar{\omega}_j^{(m)}(u \mid t), m = 1, ..., M\}$ where in each iteration step we draw $\bar{\omega}_j^{(m)}$ from $\mathcal{N}(\omega_j^{(m)}, \sigma_j^{(m)})$. We use a normal distribution since given the random effects the longitudinal outcomes are assumed to be normally distributed. Generally, this yields wider intervals compared to the credibility intervals. In summary, we get the following Monte Carlo sampling scheme

S1: Compute empirical Bayes estimate $\hat{\boldsymbol{b}}_j$.

S2: Draw $\boldsymbol{\vartheta}^{(m)}$ from $p(\boldsymbol{\vartheta} \mid \mathcal{D}_n)$.

S3: Draw $\boldsymbol{b}_j^{(m)}$ from $p(\boldsymbol{b}_j \mid T_j^* > t, \boldsymbol{y}_j(t); \boldsymbol{\vartheta})$.

S4: Compute $\omega_j^{(m)} = \boldsymbol{x}_j^\top(u)\boldsymbol{\beta}^{(m)} + \boldsymbol{z}_j^\top(u)\boldsymbol{b}_j^{(m)}$.

S5: If $u > t$: draw $\bar{\omega}_j^{(m)}$ from $\mathcal{N}(\omega_j^{(m)}, \sigma_j^{(m)})$.

S6: Repeat Steps 2-5 $m = 1, ..., M$ times.

The point-wise credibility and prediction intervals are then computed by taking the corresponding quantiles of the realizations $\{\omega_j^{(m)}(u \mid t), m = 1, ..., M\}$, $u < t$, and $\{\bar{\omega}_j^{(m)}(u \mid t), m = 1, ..., M\}$, $u > t$, respectively.

4.3 Predictive accuracy of dynamic prediction

When validating and comparing the predictive performance of the above introduced estimators the interest primarily lies in how well the longitudinal marker predicts the survival outcome. Therefore, to assess this quality, two popular measures exist in the literature of time-to-event models: discrimination and calibration. Specifically, discrimination measures how well a model can discriminate patients who will have the event from patients who will not (Pencina et al., 2008), whereas calibration is how well the model predicts the observed data (Schemper and Henderson, 2000). In the following, we present discrimination and calibration measures suitable for the previously presented dynamic prediction setting.

4.3.1 Discrimination

In models with a binary outcome, like the occurrence of an event or not, a well established methodology is to compare the survival probabilities between the subjects who are going to experience the event within a relevant time frame to those who are still event-free. To put this more formally, following Rizopoulos et al. (2017), based on a joint model and the methodology presented in section 4.1, we are interested in comparing the survival predictions $\pi_i(t + \Delta t \mid t)$ and $\pi_j(t + \Delta t \mid t)$ for a randomly selected pair of subjects $\{i, j\}$ that both provide a set of longitudinal measurements until time point t. The discriminative ability of the model can then be assessed by the area under the receiver operating curve (AUC) which equals

$$\begin{aligned}&\text{AUC}(t, \Delta t)\\ &\quad = \Pr\left[\pi_i(t + \Delta t \mid t) < \pi_j(t + \Delta t \mid t) \mid \{T_i^* \in (t, t + \Delta t]\} \cap \{T_j^* > t + \Delta t\}\right], \end{aligned} \tag{4.14}$$

that is, if subject i experiences the event in the time frame $(t, t + \Delta t]$ and subject j only at a later point, we would expect subject j to have a higher probability of still being event-free after $t + \Delta t$ compared to subject i. A weighted average of the AUCs can be used to summarize the discriminative performance of a model over the whole follow-up period. More specifically, following an approach similar to Antolini et al. (2005) and Heagerty and Zheng (2005), we can compute a weighted average of AUCs (Rizopoulos, 2012) as

$$\text{C}_{\text{dyn}}^{\Delta t} = \int_0^\infty \text{AUC}(t, \Delta t)\Pr\{\mathcal{E}(t)\}dt \Big/ \int_0^\infty \Pr\{\mathcal{E}(t)\}dt, \tag{4.15}$$

where $\mathcal{E}(t) = \left[\{T_i^* \in (t, t + \Delta t]\} \cap \{T_j^* > t + \Delta t\}\right]$, and $\Pr\{\mathcal{E}(t)\}$ denotes the probability that a randomly chosen pair is comparable at time point t. A random pair $\{i, j\}$ is comparable if their observed event times can be ordered such that subject i experiences the event in $(t, t + \Delta t]$ and subject j is known to survive longer. $\text{C}_{\text{dyn}}^{\Delta t}$ can be called a dynamic concordance index since it summarizes the weighted concordance probabilities over the whole follow-up period.

The estimation of (4.14) and (4.15) is complicated by two issues: the calculation of the integrals in the definition of $\text{C}_{\text{dyn}}^{\Delta t}$ and we need to account for censoring when comparing two random subjects. The former issue can be solved using Gaussian quadrature. In order to also consider censored subjects in the calculation of the AUC, Rizopoulos et al. (2017) suggest the following decomposition

$$\text{AUC}(t, \Delta t) = \text{AUC}_1(t, \Delta t) + \text{AUC}_2(t, \Delta t) + \text{AUC}_3(t, \Delta t) + \text{AUC}_4(t, \Delta t), \tag{4.16}$$

where the first term refers to the pairs of subjects that are comparable. Formally, we have the set

$$\Omega_{ij}^{(1)} = \left[\{T_i \in (t, t + \Delta t]\} \cap \{\delta_i = 1\}\right] \cap \{T_j > t + \Delta t\}, \tag{4.17}$$

with $i, j = 1, ..., n,\ i \neq j$. For such comparable subjects, we can estimate $\text{AUC}_1(t, \Delta t)$ as the proportion of concordant pairs out of the set of all comparable pairs

$$\hat{\text{AUC}}_1(t, \Delta t) = \frac{\sum_{i=1}^n \sum_{j=1;j\neq i}^n I\big\{\hat{\pi}_i(t + \Delta t \mid t) < \hat{\pi}_j(t + \Delta t \mid t)\big\} \times I\big\{\Omega_{ij}^{(1)}(t)\big\}}{\sum_{i=1}^n \sum_{j=1;j\neq i}^n I\big\{\Omega_{ij}^{(1)}(t)\big\}}, \tag{4.18}$$

where $I(.)$ denotes the indicator function, and $\hat{\pi}_i(.)$ and $\hat{\pi}_j(.)$ are estimated using the methodology from section 4.1. The other terms in the decomposition of (4.16) refer to those pairs whose observed

survival times T_i and T_j cannot be ordered due to censoring. Therefore we have the following three cases

$$\begin{aligned}
\Omega_{ij}^{(2)} &= \big[\{T_i \in (t, t+\Delta t]\} \cap \{\delta_i = 0\}\big] \cap \big[T_j > t + \Delta t\big], \\
\Omega_{ij}^{(3)} &= \big[\{T_i \in (t, t+\Delta t]\} \cap \{\delta_i = 1\}\big] \cap \big[\{t < T_j \leq t + \Delta t\} \cap \{\delta_j = 0\}\big], \\
\Omega_{ij}^{(4)} &= \big[\{T_i \in (t, t+\Delta t]\} \cap \{\delta_i = 0\}\big] \cap \big[\{t < T_j \leq t + \Delta t\} \cap \{\delta_j = 0\}\big],
\end{aligned}$$

with again $i, j = 1, ..., n$, $i \neq j$. In the set $\Omega_{ij}^{(2)}$ we have the situation that subject i is censored in the interval $(t, t + \Delta t]$ and the other one survives at least until $t + \Delta t$ meaning that we do not know if $T_i^* < T_j^*$. The third case $\Omega_{ij}^{(3)}$ is similar to the previous one but now subject i is known to experience the event and subject j is censored in the interval. The last set $\Omega_{ij}^{(4)}$ contains all pairs where both subjects are censored during the interval. The corresponding $\mathrm{AUC}_m(t, \Delta t)$, $m = 2, 3, 4$, can be estimated similar to (4.18) where the concordant pairs are now weighted by $\hat{\nu}_{ij}^{(m)}$ the probability that a pair $\{i, j\}$ is comparable, that is,

$$\hat{\mathrm{AUC}}_m(t, \Delta t) = \frac{\sum_{i=1}^n \sum_{j=1; j\neq i}^n I\big\{\hat{\pi}_i(t + \Delta t \mid t) < \hat{\pi}_j(t + \Delta t \mid t)\big\} \times I\big\{\Omega_{ij}^{(m)}(t)\big\} \times \hat{\nu}_{ij}^{(m)}}{\sum_{i=1}^n \sum_{j=1; j\neq i}^n I\big\{\Omega_{ij}^{(1)}(t)\big\}}, \tag{4.19}$$

In detail, we have $\hat{\nu}_{ij}^{(2)} = 1 - \hat{\pi}_i(t + \Delta t \mid T_i)$ denoting the probability that the censored subject i experiences the event before $t + \Delta t$; $\hat{\nu}_{ij}^{(3)} = \hat{\pi}_j(t + \Delta t \mid T_j)$ is the probability that subject j whose censoring time lies in the interval is actually event free at least until $t+\Delta t$; and $\hat{\nu}_{ij}^{(4)} = \{1-\hat{\pi}_i(t+\Delta t \mid T_i)\} \times \hat{\pi}_j(t + \Delta t \mid T_j)$ being the probability that the event occurs in the interval for subject i but not for subject j.

Having an estimate for $\mathrm{AUC}(t, \Delta t)$, we are able to estimate $\mathrm{C}_{\mathrm{dyn}}^{\Delta t}$. The first step is computing the weights $\Pr\{\mathcal{E}(t)\}$ which can be rewritten as

$$\begin{aligned}
\Pr\{\mathcal{E}(t)\} &= \Pr\big[\{T_i^* \in (t, t+\Delta t]\} \cap \{T_j^* > t + \Delta t\}\big] \\
&= \Pr(T_i^* \in (t, t+\Delta t]) \times \Pr(T_j^* > t + \Delta t) \\
&= \{S(t) - S(t + \Delta t)\} S(t + \Delta t)
\end{aligned} \tag{4.20}$$

where we assume independence of subject i and j, and $S(.)$ being the marginal survival function (see section 2.2.1). From that, we can obtain an estimate $\hat{\Pr}\{\mathcal{E}(t)\} = \{\hat{S}(t) - \hat{S}(t + \Delta t)\}\hat{S}(t + \Delta t)$, with $\hat{S}(.)$ denoting the Kaplan-Meier estimate of the marginal survival function which is based on the new data. Combining this estimate with the estimation of $\mathrm{AUC}(t, \Delta t)$, we can obtain an estimate for the dynamic concordance index

$$\hat{\mathrm{C}}_{\mathrm{dyn}}^{\Delta t} = \frac{\sum_{k=1}^K w_k \hat{\mathrm{AUC}}(t_k, \Delta t) \times \hat{\Pr}\{\mathcal{E}(t_k)\}}{\sum_{k=1}^K w_k \hat{\Pr}\{\mathcal{E}(t_k)\}}, \tag{4.21}$$

where t_k and w_k, $k = 1, ..., K$, denote the corresponding nodes and weights for a K-point Gaussian quadrature rule on the interval $[0, t_{\max}]$, respectively.

4.3.2 Calibration

The assessment of how well the assumed model predicts the observed data is usually based on the expected error of predicting future events. In our setting, we are specifically interested in the

accuracy of predicting the occurrence of events at u given the observed information until t, where $u > t$. Therefore, following Rizopoulos (2016), a commonly used measure is the expected prediction error

$$\text{PE}(u \mid t) = E\left[L\{I(T_i^* > u) - \pi_i(u \mid t)\}\right], \tag{4.22}$$

where $I(T_i^* > u)$ denotes the event status at time point t, and $L(.)$ can be any loss function, such as absolute or square loss. The expectation is taken with respect to the distribution of the event times. Henderson et al. (2002) suggest an estimate of $\text{PE}(u \mid t)$ that also accounts for censoring, that is

$$\begin{aligned}\hat{\text{PE}}(u \mid t) = n(t)^{-1} \sum_{i:T_i \geq t} & I(T_i \geq u) L\{1 - \hat{\pi}_i(u \mid t)\} + \delta_i I(T_i < u) L\{0 - \hat{\pi}_i(u \mid t)\} \\ & + (1 - \delta_i) I(T_i < u) \left[\hat{\pi}_i(u \mid T_i) L\{1 - \hat{\pi}_i(u \mid t)\} + \{1 - \hat{\pi}_i(u \mid T_i)\} L\{0 - \hat{\pi}_i(u \mid t)\}\right],\end{aligned} \tag{4.23}$$

where $n(t)$ denotes the number of subjects observed until t. The first term of the sum corresponds to subjects that are still event-free until u, whereas the second term considers subjects whose event occur in the interval $[t, u)$; the third term takes into account the subjects censored in this interval. In order to not only measure the predictive accuracy at one specified point u given the longitudinal information until t, we can compute a weighted average of $\{\text{PE}(s \mid t), t < s < u\}$ that summarizes the error of prediction in the interval $[t, u]$ and further corrects for censoring. Such an estimator has been proposed by Schemper and Henderson (2000) and was adapted to our time dynamic setting by Rizopoulos (2016), that is,

$$\hat{\text{IPE}}(u \mid t) = \frac{\sum_{i:t \leq T_i \leq u} \delta_i \{\hat{S}_C(t) / \hat{S}_C(T_i)\} \hat{\text{PE}}(T_i \mid t)}{\sum_{i:t \leq T_i \leq u} \delta_i \{\hat{S}_C(t) / \hat{S}_C(T_i)\}}, \tag{4.24}$$

with $\hat{S}_C(.)$ denoting the Kaplan-Meier estimator of the censoring time distribution.

4.4 Implementation details

The former introduced subject-specific predictions for survival probabilities, Section 4.1, and longitudinal outcomes, Section 4.2, can be estimated using the function `jm_dynpred()`. The function accepts as its two main arguments a fitted joint model based on which the predictions will be estimated and a data frame containing the data on the new subjects for which the outcomes should be predicted. The joint model needs to be a flexible Bayesian additive joint model (Section 3) fitted using the function `bamlss` from package `bamlss`. The main usage of the function is `jm_dynpred(object, newdata)` where the object `object` needs to be of class `bamlss`. Further additional arguments that may be specified by the user and a short description can be found in Table 1.

For the predicted survival and longitudinal outcomes the function `jm_dynpred` follows the algorithms introduced in Section 4.1 and 4.2. In order to be able to compute the empirical Bayes estimates $\hat{\boldsymbol{b}}$ for the new subjects (Step 1), the covariance matrix of the random effects $\boldsymbol{D}$ is estimated using the estimates for the random effects from the provided joint model. Steps 2 and 3 are implemented as described in the algorithms. The Metropolis-Hastings algorithm in Step 3 is run externally which allows to determine the number of iterations, as well as potential burn-in and thinning parameters which is done by specifying the corresponding arguments `n.iter`, `burnin` and `thin`, respectively. The default arguments result in a Metropolis-Hastings algorithm with M iterations, the number of Monte Carlo samples, and no burn-in phase or thinning. The user can further set the argument `scale` which influences the acceptance rate of the Metropolis-Hastings algorithm by scaling the variance of the proposal density by the supplied factor, meaning that a higher factor and thus a larger variance decreases the acceptance rate. Note that a requirement for the use of the

Name	*Short description*
`object`	a fitted joined model of class `bamlss`
`newdata`	data frame that contains new data
`survtimes`	numeric vector that contains specific time points for prediction
`last.time`	numeric vector or variable name containing the last time points at which subjects in `newdata` were known to be event-free
`nGL`	scalar denoting the number of points for Gauss-Legendre Quadrature
`simulate`	logic; if `TRUE` Monte Carlo estimator will be used; if `FALSE` simple estimator without credibility intervals is used
`M`	integer denoting number of Monte Carlo samples
`CI.levels`	numeric containing the two quantiles for the credibility intervals
`scale`	scalar that controls acceptance rate of Metropolis-Hastings algorithm
`n.iter`	integer denoting the number of iterations in Metropolis-Hastings algorithm
`burnin`	integer denoting length of burn-in phase
`thin`	integer denoting thinning parameter

Table 1: Arguments that may be specified in function `jm_dynpred()`. First six are general arguments, whereas the latter are only needed when using Monte Carlo estimator, including Metropolis-Hastings algorithm.

estimators that are based on Monte Carlo sampling is that the original model was fitted via MCMC sampling. If this is not the case there exists no sample to sample from in Step 2 and the function thus uses the first-order estimators (4.4) and (4.11) that however do not provide any credibility intervals. Using the argument `simulate = FALSE` yields these estimates as well. For all integrals we use a Gauss-Legendre quadrature rule where the corresponding nodes and weights are obtained from the function `gaussLegendre()` of package `pracma` where the default are 15 points.

In summary, there are three different ways of estimating the dynamic prediction: (1) the first-order estimator which does not provide any credibility intervals but therefore it is computationally faster and does not require an existing MCMC sample from the joint model, (2) the estimator introduced by Rizopoulos (2016) that is based on a Monte Carlo simulation and only draws in total M samples in the Metropolis-Hastings algorithm for the random effects which does not necessarily yield a converged Markov chain, and (3) the Monte Carlo estimator with an externally run Metropolis-Hastings algorithm. The estimators are used when setting the following arguments in the function call: (1) `simulate = FALSE`, (2) default estimator if MCMC sample is provided by joint model, (3) `n.iter` and if desired also `burnin`, `thin`.

The resulting `R` object is of class `dynamicpred.bamlss`. This object is a list containing either seven or ten components depending on the type of estimator used for the prediction. An overview of the returned values is presented and explained in Table 2.

To examine the resulting dynamic prediction two methods, `print()` and `plot()`, for class `dynamicpred.bamlss` are implemented. The `print` method simply prints the list `summaries` that includes the prediction time points, the predicted survival probabilities and if available the corresponding credibility intervals, for each subject. The default `plot` method produces a plot for one subject that is divided into two parts (see for instance Figure 5). The upper part plots a line for the predicted survival probabilities, whereas the lower part shows the longitudinal trajectory of the biomarker: first during the observation time frame and then marked by a vertical line the longitudinal predictions. If available, the corresponding point-wise credibility intervals are presented as well.

Moreover, the user can plot the traceplots and autocorrelation functions for the coefficients of the random effects from the Metropolis-Hastings algorithm by setting the argument `which` to `"samples"`.

Name	*Short description*
`summaries`	list that contains predicted survival probabilities and if available corresponding credibility intervals for each subject
`survtimes`	time points used for prediction
`last.time`	numeric with last time points where subjects were known to be event-free
`obs.times`	list that contains observation time points for each subject
`y`	possibly transformed observed marker values
`modes.b`	empirical Bayes estimates for random effects
`y.long`	list that contains longitudinal outcomes during and after last observed time point and if available corresponding credibility intervals for each subject
`full.results`	list containing all Monte Carlo samples
`success.rate`	matrix where each column contains acceptances of the Metropolis-Hastings algorithm for one subject
`b`	list containing all Metropolis-Hastings samples

Table 2: Components of a `dynamicpred.bamlss` object. First seven components result for first-order estimator. Additional components result for Monte Carlo sampling.

In order to evaluate the quality of the dynamic prediction all previously introduced measures are as well implemented. In detail, for the discrimination the estimates for $\text{AUC}(t, \Delta t)$ (4.14), and its weighted average $\text{C}_{\text{dyn}}^{\Delta t}$ (4.15), can be obtained using the functions `jm_auc()` and `jm_dynC()`. Both functions require as its main arguments a fitted joint model of class `bamlss` and a data frame with the data that should be used for the evaluation. Further, to use the function `jm_auc()` the user has to specify t and Δt via the arguments `Tstart` and `dt`. For the use of `jm_dynC()` only the length of the time interval for the prediction Δt needs to be set using the argument `dt`. For the calibration measures, the prediction error $\text{PE}(u \mid t)$ (4.23) can be estimated using the function `jm_prederr()`. Again its two main arguments are a fitted joint model of class `bamlss` and a data frame and further the prediction interval $[t, u]$ needs to be stated using the arguments `Tstart` for the starting time point and `Thoriz` denoting the ending point. The integrated prediction error $\text{IPE}(u \mid t)$ (4.24) is computed by setting the argument `interval = TRUE`. Note that the default loss function is a squared loss function but any other loss function can be defined using the argument `lossFun`. In all 4 methods the default for the dynamic prediction are 100 Monte Carlo iterations including a Metropolis-Hastings algorithm with 100 iterations and neither any burn-in nor any thinning.

5 Analysis of PBC data

In this Section we are going to apply the dynamic prediction for flexible Bayesian additive joint models on the PBC data set which is widely used in the joint modeling framework. The data is available in the R package `JMbayes` and was collected as part of a study on primary biliary cirrhosis (PBC) which is a fatal, rare liver disease. The focus is going to lie in applying our implemented framework but also to demonstrate the use of the different functions and to compare our results to the ones obtained from a similar model fit in package `JMbayes`. `JMbayes` is the standard R-package used for analyzing joint models. Compared to `bamlss` it allows a less flexible specification of the subject-specific trajectories and non-linear effects.

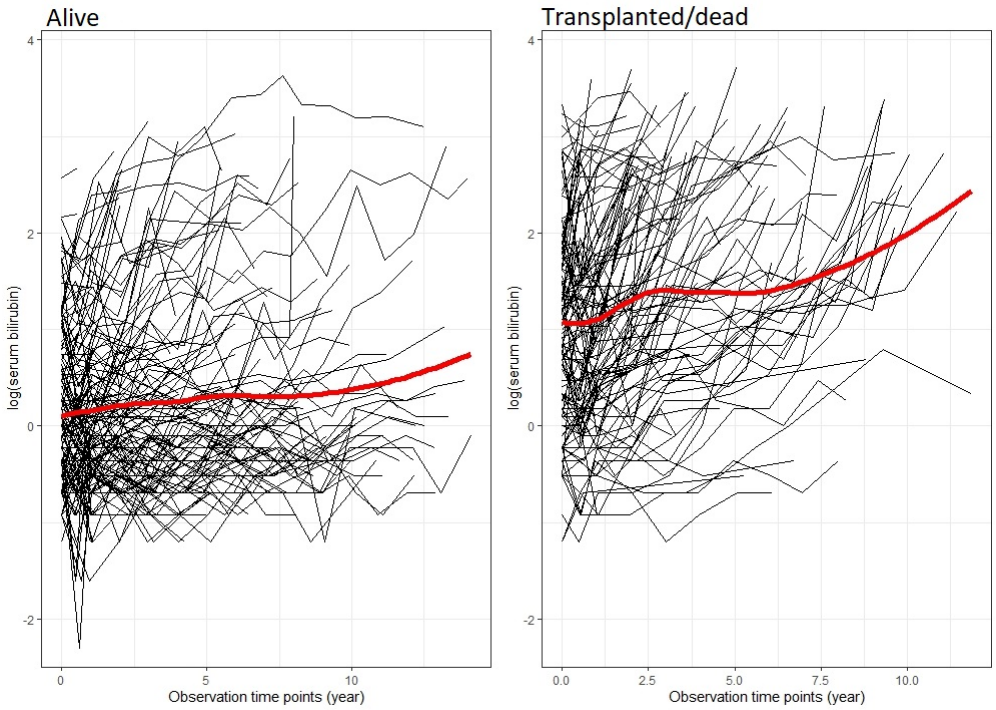

Figure 2: Longitudinal trajectories for log serum bilirubin for subjects that were alive (left) or had a transplantation/died (right). Red lines are smooth function (LOESS) for the corresponding trajectories.

5.1 Data set

Primary biliary cirrhosis (PBC) is a fatal but rare liver disease which results from a destruction of small bile ducts within the liver. From 1974 to 1984 Mayo Clinic conducted a study on PBC where they compared the impact of the drug D-penicillamine with a placebo. The provided data frame consists of 312 subjects, 158 randomized to D-penicillamine and 154 to placebo, that were followed over a period up to ten years (Fleming and Harrington, 2011). By the end of the study 140 (45%) patients died, 29 (9%) received a transplant and 143 (46%) were still alive. The data set `pbc2` that is available in the R package `JMbayes` provides information on several baseline covariates, such as age and sex, as well as follow-up measurements on some biomarkers. The biomarker we will focus our analysis on is the serum bilirubin level which was shown to be associated with the progression of the disease (Rizopoulos, 2016). Each patient was on average observed 6.2 (standard deviation 3.8) times which leads in total to 1945 observations of serum bilirubin where the visits where scheduled at 6 months, 12 months and annually therafter. The trajectories of log serum bilirubin for five randomly selected subjects which are presented in Figure 1 emphasize the need for a nonlinear and subject-specific modeling of the biomarker. Figure 2 strongly indicates that a higher serum bilirubin level raises the probability for a transplant or death.

Figure 3 presents a descriptive plot for the survival outcomes. More specifically, we present the Kaplan-Meier estimates for the two groups that were prescribed different drugs. In the plot there is no clear tendency that the drug D-penicillamine improves the disease progression.

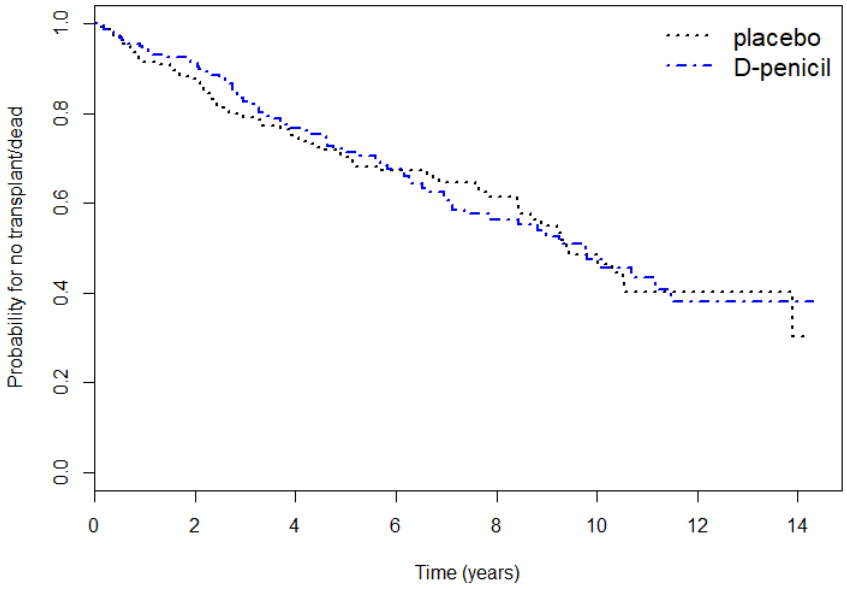

Figure 3: Kaplan Meier estimates for survival outcomes in PBC data. Blue line represents estimates for the group that was prescribed to D-penicillamine, black line represents placebo group.

5.2 Model fit

First we are going to fit a joint model using the package `bamlss`. Therefore, we model the hazard together with a time-varying association between the log serum bilirubin and the hazard, assume a flexible baseline hazard and a constant error variance in the longitudinal submodel. The longitudinal submodel is modeled as described in (3.16) where the smooth effect of time (`year`) and the functional random intercepts are modeled using P-splines with cubic B-splines, second order differences and 12 knots (4 internal knots) yielding 7 basis functions after the application of the sum-to-zero constraints. The predictors η_λ and η_α are modeled as a smooth function of the observed survival times (`years`) using P-splines with 14 knots (6 internal knots) which results in 9 basis functions after the application of constraints. The baseline covariates `drug` and `age` are modeled as binary and linear effects, respectively. To fit a model in `bamlss` all predictors are given to the function call in form of a list.

```
library(bamlss)
library(JMbayes)
data(pbc2)

f <- list(
  Surv2(years, status2, obs = log(serBilir)) ~ s(years, k = 10, bs = "ps"),
  gamma ~ drug + age,
  mu ~ ti(id, bs = "re") + ti(year, bs = "ps", k = 8) +
       ti(id, year, bs=c("re", "ps"), k=c(nlevels(pbc2$id), 8)),
  sigma ~ 1,
  alpha ~ s(years, k=10, bs="ps"),
  dalpha ~ -1
```

```
)

set.seed(938475)
joint_model <- bamlss(f, data = pbc2, family = "jm", timevar = "year",
                      idvar = "id", maxit = 300, n.iter = 23000, burnin = 3000,
                      thin = 20)
```

The function `bamlss()` first estimates the posterior mode and uses these estimates as starting values for the sampler. In our specification 23,000 samples are drawn where the first 3,000 are discarded and we only keep every 20th sample leading in total to 1,000 samples. The estimation of the model is quite lengthy and takes approximately 3 days on a single core of a 3.40 GHz Intel Xeon Processor E5-2643. Note that on a Linux server the estimation of the joint model can be parallelized by using multiple cores via the argument `cores`. Then each core starts its own chain to reduce computation time (burn-in and thinning parameters are applied to each chain). To check the convergence and mixing of the sampler we can inspect traceplots and the corresponding autocorrelation functions that are presented in Figure 17 in the Appendix.

To further examine the estimated association between the biomarker and the event, we plot the predictor $\eta_\alpha(t)$ using predicted values of the effect at the observed event times that are obtained from the function `predict()`. In Figure 4 we see a positive, but rather constant and in the end a little decreasing effect which indicates that a higher serum bilirubin level increases on average the hazard rate and therefore the risk of death.

```
pred_alpha <- predict(joint_model, model="alpha", newdata=pbc2.id, FUN=c95)

plot2d(pred_alpha ~ pbc2.id[, "years"], rug = TRUE)
abline(h = 0, lty = 2)
```

Having a look at the summary of the estimated baseline coefficients, we see that prescribing the drug D-penicillin instead of a placebo decreases the hazard rate on average by a multiplicative factor of $\exp(-0.04) = 0.96$ but it is important to point out that this effect is not significant according to 95% credibility intervals which is consistent with the result from the Kaplan Meier estimates. Moreover, a higher age increases the hazard on average by the factor $\exp(0.066) = 1.07$ which is also what one would expect. Note, that in `bamlss` the estimated intercept for predictor η_γ serves as a joint intercept for the predictors η_γ and η_λ and should therefore not be interpreted from an essential point of view.

```
# Formula gamma:
# ---
# gamma ~ drug + age
# -
# Parametric coefficients:
#                     Mean     2.5%      50%    97.5%
# (Intercept)    -7.89336 -9.03915 -7.87592 -6.74849
# drugD-penicil  -0.04300 -0.37865 -0.03994  0.29802
# age             0.06593  0.04826  0.06583  0.08395
# -
```

Further model results including plots for the baseline hazard as well as for the longitudinal predictor η_μ are presented in the Appendix C.1.

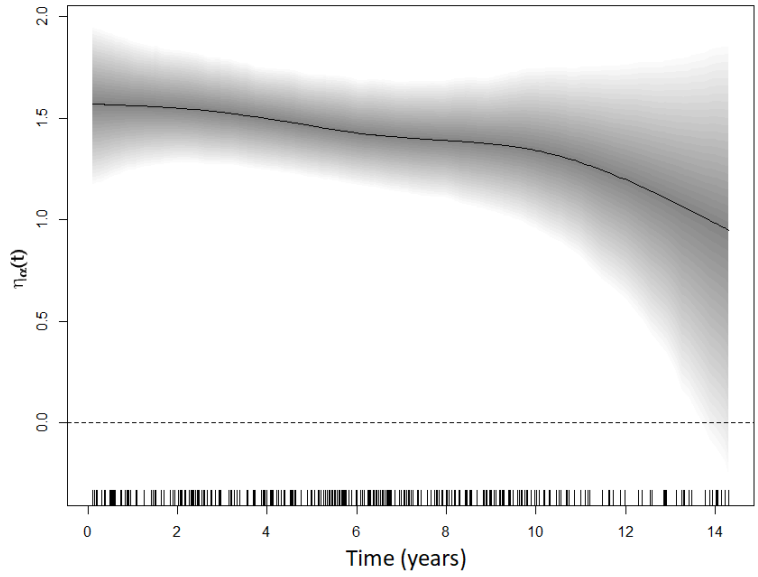

Figure 4: Mean estimates and 95% pointwise credibility intervals for the association $\eta_\alpha(t)$ between log(serum bilirubin) and hazard from model fit. Rugs at bottom indicate observed event times.

5.3 Dynamic prediction

Based on the fitted joint model, we will now estimate in-sample subject-specific dynamic predictions for the survival and longitudinal outcomes for two subjects to illustrate how changes in the biomarker affect the conditional survival probabilities. One subject has a constant serum bilirubin level (ID 83) and the other one has an increasing level (ID 21). Note that this is an in-sample prediction since both subjects are also used for fitting the joint model. First, we generate a new data frame that only contains the observations on those two subjects 21 and 83. The dynamic predictions are estimated using the function `jm_dynamicpred()` where we specify the prediction time points via the argument `survtimes`. The longitudinal observations in the PBC data range from 0 to 14.1. We are however predicting until time point 20. In this case the spline basis for all smooth predictors are linearly extrapolated. To investigate the results, we use the implemented `plot` method. The output for both subjects is shown in Figure 5. We clearly see that an increasing longitudinal trajectory leads to a faster decrease in the conditional survival probabilities. Moreover, as expected, the credibility intervals become larger for longer prediction intervals.

```
ND <- pbc2[pbc2$id %in% c(21, 83), ]
dynpred <- jm_dynamicpred(joint_model, newdata = ND,
                          survtimes = seq(0, 20, by = 0.5))

plot(dynpred, id = 21)
plot(dynpred, id = 83)
```

In the following we demonstrate the dynamics of the prediction by plotting the estimated dynamic predictions for patient 21 at several time points. Specifically, we generate four different plots after 0, 1, 3 and 4.9 years which is done using a `for`-loop:

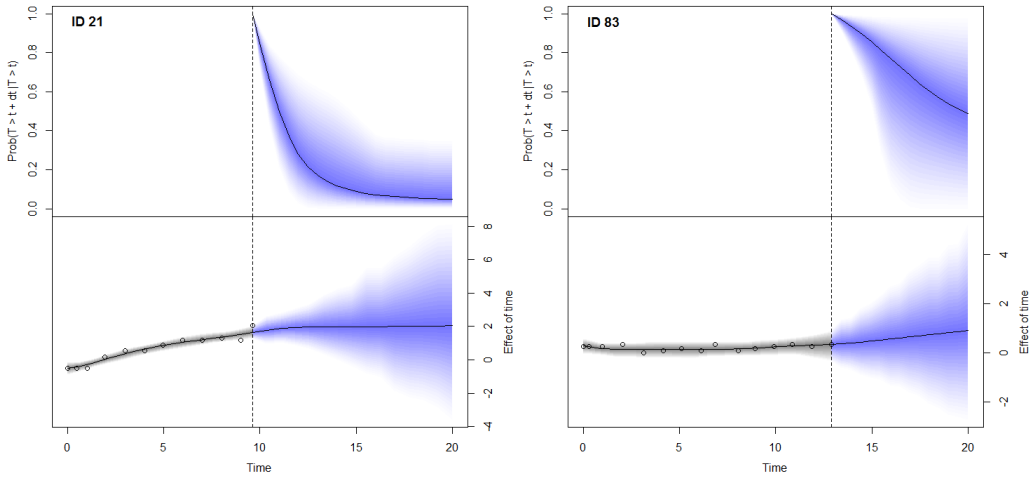

Figure 5: Dynamic predictions for the survival (upper plots) and longitudinal (lower plots) outcome for Patients 21 (left) and 83 (right) from *pbc2* data. Vertical dotted lines indicate time point of last observation. Solid lines are mean estimates and shaded area around are 95% credibility intervals (grey) and prediction intervals (blue).

```
ND <- pbc2[pbc2$id == 21, ]
dynpred_21 <- vector("list", length=nrow(ND))

for(i in 1:nrow(ND)) {
  dynpred_21[[i]] <- jm_dynamicpred(joint_model, newdata = ND[1:i, ])
}

for(i in c(1, 3, 5, 7)) {
  plot(dynpred_21[[i]], estimator = "median")
  title(main = paste("Follow-up time:", round(ND$year[i], 1), sep = " "),
  outer = TRUE)
}
```

The four plots are shown in Figure 6 where in this case the solid lines represent the median estimates which is specified via the argument `estimator`. We observe that after the third measurement there is an increase in the serum bilirubin level and at the same time the line for the conditional survival probabilities becomes more steep. Comparing the predicted survival probabilities at time point 11 we further see that in the two latter plots a lower survival probability is predicted although the subject was longer known to be event free.

Moreover, we can check the convergence of the random effects coefficients that are sampled using a Metropolis-Hastings algorithm by setting the argument `which = "samples"` in the call to the `plot` function.

```
plot(dynpred, id = 21, which = "samples")
```

The output for the first 4 coefficients for patient 21 are presented in Figure 7. We see that there is almost no autocorrelation in the chains but looking at the traceplots there are small fluctuations

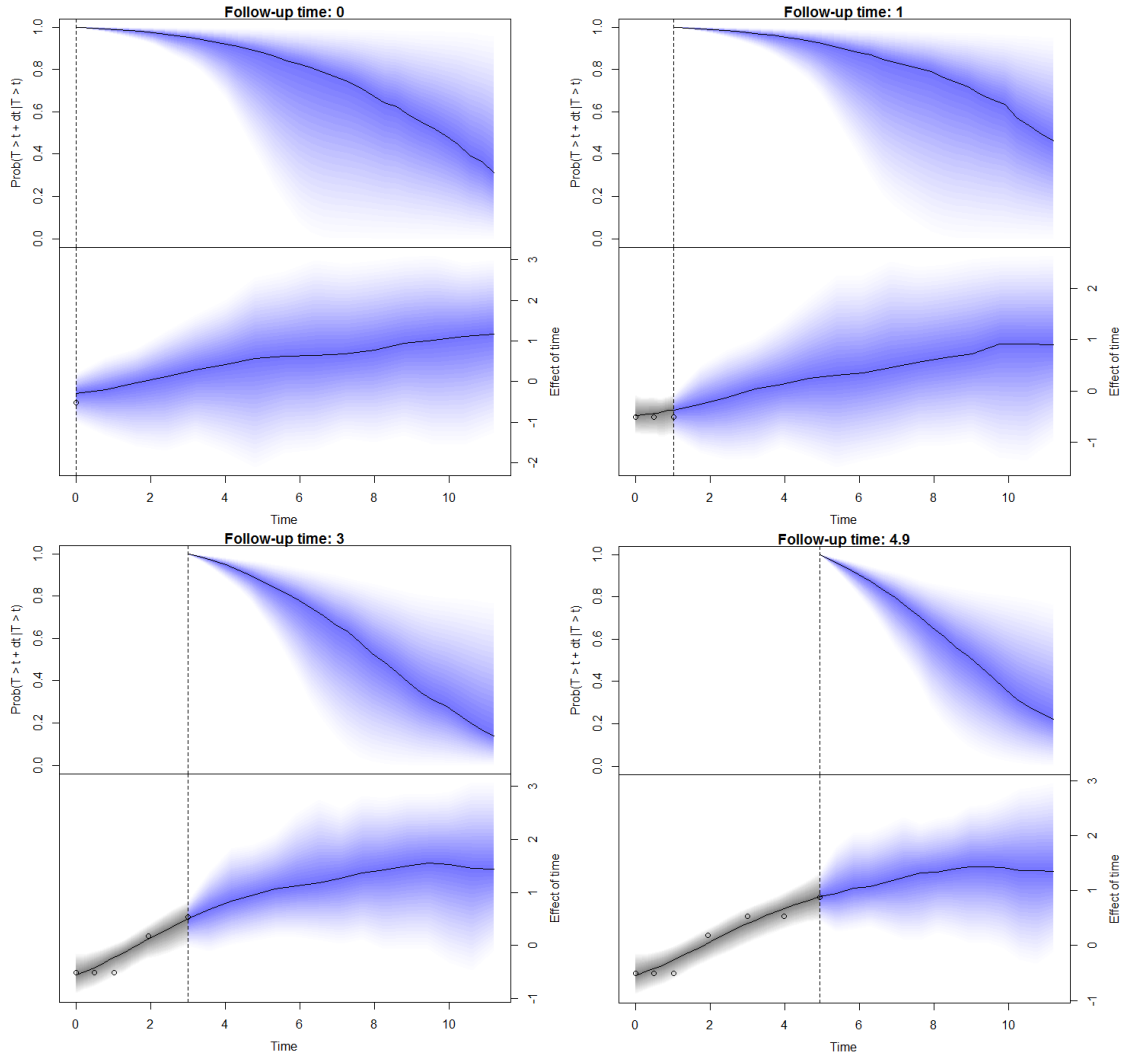

Figure 6: Dynamic predictions for the survival (upper plots) and longitudinal (lower plots) outcome for Patient 21 from *pbc2* data. The vertical dotted lines indicate the time point of the last observation. The solid lines are the median estimates and the shaded area around are 95% credibility intervals (grey) and prediction intervals (blue).

that could be weakened by extending the chain and including an additional burn-in phase. Thus, we estimate the dynamic predictions again this time using 2000 iterations in the Metropolis-Hastings algorithm together with a burn-in of 700. The results for this setting are presented in Appendix Figure 20. The traceplots are now more constant. However, when comparing the dynamic predictions from both approaches (Figures 5 and 21) there is no essential change which encourages the presumption that the empirical Bayes estimates for the random effects are good starting values for the sampler and it is thus sufficient to base the dynamic predictions in this case on a Metropolis-Hastings algorithm with only 100 iterations.

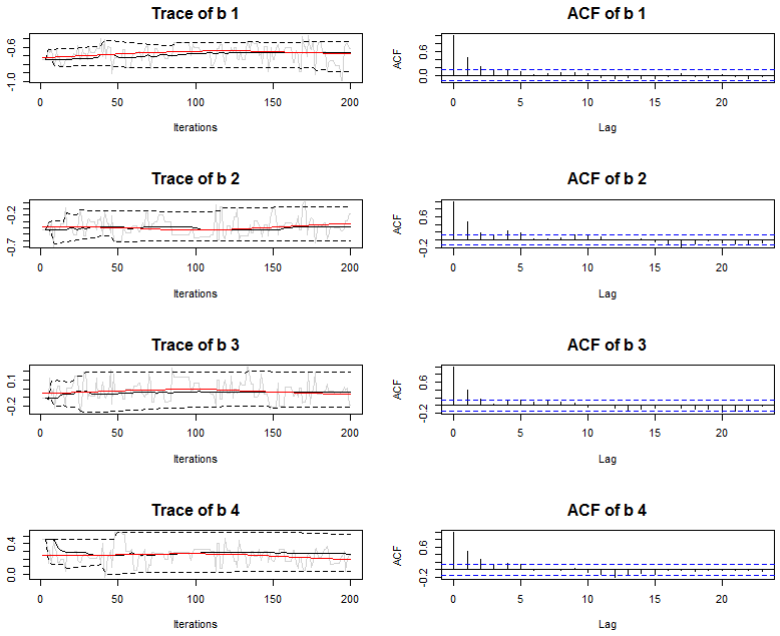

Figure 7: Traceplots (left) and autocorrelation functions (right) for the first 4 random effects coefficients of patient 21 sampled using a Metropolis-Hastings algorithm. Red lines indicate LOESS smoother; black lines running median.

5.4 Evaluation

The quality of the dynamic prediction can be assessed by means of calibration and discrimination measures that were previously introduced. To evaluate the ability of the model to discriminate between patients that are going to have the event in a relevant time frame and those being still event free, we can use the function `jm_auc()`, or its dynamic version `jm_dynC()`.

```
auc_pbc <- jm_auc(joint_model, newdata=pbc2, Tstart=5, Thoriz=7)
auc_pbc

#          Time-dependent AUC for the Joint Model joint_model
#
# Estimated AUC: 0.8669
```

```
# At time:  7
# Using information up to time:  5 (202 subjects still at risk)

dynC_pbc <- jm_dynC(joint_model, newdata=pbc2, dt=2)
dynC_pbc

#         Dynamic Discrimination Index for the Joint Model joint_model
#
# Estimated dynC: 0.8751
# In the time interval:  [0, 14.3057]
# Length of time interval:  2
```

From the output we see that the fitted model is well suitable to discriminate patients who are going to die within the next two years from patients that are still alive, taking into account all available information from the first five years. This is also the case for prediction intervals of length two for the whole follow-up period which is indicated by a dynamic C index of 0.88.

Both the prediction error and integrated prediction error are computed via the function `jm_prederr()` with a similar syntax to the function `jm_auc()`. By default a squared loss function is used.

```
pe_pbc <- jm_prederr(joint_model, newdata=pbc2, Tstart=5, Thoriz=7)
pe_pbc

# Prediction Error for the Joint Model joint_model
#
# Estimated prediction error:  0.0801
# At time:  7
# Using information up to time:  5 (202 subjects still at risk)
# Loss function:  square
```

To compute the integrated prediction error we only set the argument `interval = TRUE`. Both measures indicate a good predictive quality.

```
ipe_pbc <- jm_prederr(joint_model, newdata=pbc2, Tstart=5, Thoriz=7, interval=TRUE)
ipe_pbc

# Prediction Error for the Joint Model joint_model
#
# Estimated prediction error:  0.0486
# In the time interval:  [5, 7]
# Using information up to time:  5 (202 subjects still at risk)
# Loss function:  square
```

Overall, we can conclude that the dynamic prediction for the above fitted joint model performs well in both evaluation measures. However, it is important to point out, that when using these measures for our fitted joint model we are confronted with the problem that the estimated predictive performance may be too overoptimistic since we are using the same data the model was fitted to. Therefore, to account for this fact, we carry out a cross validation, more accurately a 10-fold cross validation, as described in Rizopoulos (2016). First, we randomly split the *pbc2* data in 10 sub data sets.

```
f <- 10
n <- nrow(pbc2.id)
sub <- split(seq_len(n), sample(rep(seq_len(f), length.out = n)))
```

Now we fit the joint model 10 times each time leaving out a different sub data set that can be used for validating the predictive quality of the model. To speed up the computation time we parallelize the estimation using the package `parallel`. Note, that for the IPE two out of ten results yield `NA`s since for the corresponding testing data sets no actual events occur in the interval $[5, 7]$.

```
library("parallel")

cross_val <- function(i, f) {
  library(JMbayes)
  library(bamlss)
  source('dynamicpred.R')
  data(pbc2)

  training_data <- pbc2[!pbc2$id %in% i, ]
  testing_data <- pbc2[pbc2$id %in% i, ]

  joint_model_fit <- bamlss(f, data = training_data, family = "jm",
                            timevar = "year", idvar = "id",
                            n.iter = 20000, burnin=3000, thin = 15, maxit=300)

  dynC <- jm_dynC(joint_model_fit, newdata = testing_data, dt = 2)
  auc <- jm_auc(joint_model_fit, newdata=testing_data, Tstart=5, Thoriz=7)
  ipe <- jm_prederr(joint_model_fit, newdata = testing_data, Tstart = 5,
                  Thoriz = 7, interval = TRUE)
  pe <- jm_prederr(joint_model_fit, newdata = testing_data, Tstart = 5,
                   Thoriz = 7)

  list(dynC = dynC, IPE = ipe, AUC=auc, PE=pe)
}


cl <- makeCluster(10)
res <- parLapply(cl, sub, cross_val, f=f)
stopCluster(cl)


mean(sapply(res, function(x) x$AUC$auc))
[1] 0.8382341
mean(sapply(res, function(x) x$dynC$dynC))
[1] 0.8601051

mean(sapply(res, function(x) x$PE$prederr))
[1] 0.07715355
mean(sapply(res, function(x) x$IPE$prederr), na.rm=TRUE)
[1] 0.07119993
```

Comparing the results from the cross validation to the ones previously obtained, we see that our first validation was slightly too optimistic especially for the integrated prediction error, but still the dynamic prediction based on the fitted joint model seems to perform well in predicting the conditional survival probabilities.

So far, all presented evaluations are based on dynamic predictions using a Monte Carlo simulation scheme with 100 iterations. However, if the interest is not in examining the precision of the resulting estimated probabilities we can also use the first-order estimators (4.4) and (4.11) that yield faster computation times which is especially beneficial for the estimation of the dynamic C index and the integrated prediction error. Moreover, those estimators can be used to compute dynamic predictions for joint models that were only fitted via posterior mode estimation. Table 3 presents a comparison of the in-sample evaluation as well as the 10-fold cross validation for all 4 evaluation measures and its corresponding computation times.

	In-sample [5, 7]				Cross-Validation [5, 7]			
Estimator	$C_{\text{dyn}}^{\Delta t=2}$	AUC	IPE	PE	$C_{\text{dyn}}^{\Delta t=2}$	AUC	IPE	PE
First-order	0.878	0.867	0.049	0.079	0.864	0.837	0.071	0.078
	(17.1)	(2.2)	(27.8)	(1.3)	(1.53)	(0.12)	(0.35)	(0.11)
Monte Carlo	0.875	0.867	0.049	0.080	0.860	0.838	0.071	0.077
	(18.8)	(3.4)	(42.7)	(2.1)	(2.1)	(0.14)	(0.5)	(0.17)

Table 3: Comparison of first-order estimator and Monte Carlo simulation scheme by means of discrimination (C_{dyn} and AUC) and calibration (IPE and PE) measures using all available information until year five (for computation of AUC, IPE and PE) and a prediction interval of length 2. Numbers in parenthesis indicate computation time in minutes.

Comparing both approaches, we get an almost identical performance for all evaluation measures. The reason for these results is that the first-order estimators are based on the empirical Bayes estimates for the random effects and the coefficient estimates provided by the joint model fit. As Figure 7 already indicates we can assume that the empirical Bayes estimates are already satisfying estimates for the random effects. Therefore, both estimation approaches yield very similar point estimates and thus almost identical performance measures. Moreover, the computation time decreases for all measures when using the first-order estimators instead of Monte Carlo sampling.

5.5 Comparison to JMbayes

Before being able to compare our above presented results to the dynamic prediciton implemented in `JMbayes` we need to fit a similar joint model in `JMbayes`. The corresponding longitudinal submodel is fitted with a smooth effect of time as well as random intercepts and subject-specific deviations from the overall smooth time effect. All smooth effects are based on B-splines with no internal knot yielding 3 basis functions that were chosen to minimize the DIC. The baseline hazard is modeled using P-splines with 9 basis functions. Further, we assume a constant association of the biomarker and the hazard. Note, that our joint model fit in `bamlss` so far models a time-varying association. To be better able to compare both approaches we will later in this Section refit the `bamlss` joint model assuming a constant predictor η_α. For the estimation we use the default arguments meaning 20,000 iterations with a burn-in of 3,000 and thinning which leaves 2,000 MCMC samples. The benefit of this model compared to `bamlss` is clearly its estimation time which is in this case roughly 2 minutes on a single core of a 3.40 GHz Intel Xeon Processor E5-2643. In Table 4 we compare

the resulting baseline survival covariates as well as the estimates for the intercepts for predictor η_α. Both packages yield very similar results indicating that prescribing the drug D-penicillamine instead of a placebo has no significant effect on the hazard rate and a higher age increases the hazard function on average by the factor 1.07 per year. Moreover, both packages estimate a clearly positive association between the log serum bilirubin level and the risk of death. Further plots presenting the smooth effects in the longitudinal submodel are shown in the Appendix Figure 22.

	Posterior mean coefficient estimates		
	η_γ		η_α
Package	D-penicil	Age	Intercept
`bamlss`	-0.043	0.066*	1.412*
`JMbayes`	-0.059	0.066*	1.470*

* indicates that 0 is not contained in 95% credibility intervals.

Table 4: Comparison of selected posterior mean coefficient estimates from fitted joint models in `bamlss` and `JMbayes`.

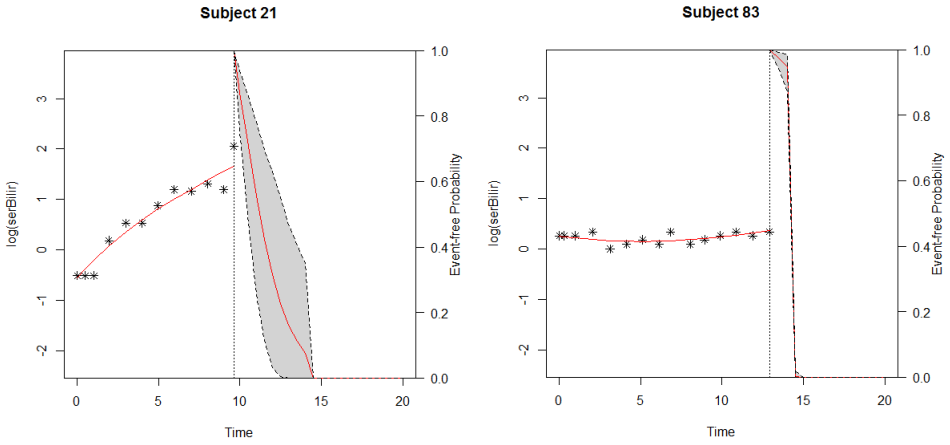

Figure 8: Dynamic Prediction for survival probabilities (right) and longitudinal trajectory (left) based on joint model fitted in `JMbayes` for patients 21 and 83 from PBC data. Red lines indicate mean estimates and shaded areas are 95% credibility intervals.

To compare the results of the dynamic predictions from both packages we are first going to reproduce Figure 5 in package `JMbayes` using the above fitted model. The results for patients 21 and 83 are presented in Figure 8 where we directly see that the dynamic prediction in `JMbayes` fails to predict values outside of the range for the observation time points used in the model fit. This happens because the basis functions for all smooth effects become quickly zero outside of this range and are not linearly extrapolated as for example in `bamlss`. Therefore, to be better able to compare the outputs we reproduce this plot in both packages only using the available information of the first 8 years. The resulting Figure 9 yields very similar courses of the survival probabilities and credibility intervals in both packages. Furthermore, in both approaches the survival predictions for patient 21 who has an increasing log serum bilirubin level decrease faster compared to patient

83. Unfortunately, we cannot directly compare the predicted longitudinal trajectories since they are not provided in `JMbayes` when computing the predicted survival probabilities. The computation and plotting for the longitudinal prediction in `JMbayes` is a little laborious since there exists no corresponding `plot` method. Moreover, the results are quite misleading since although the estimation is based on 200 Monte Carlo iterations it only returns the first-order estimate (4.11) as a sort of mean estimate. Nevertheless, Figure 23 in the Appendix presents the predicted log serum bilirubin levels for both patients 21 and 83 obtained from `JMbayes`. In contrast to the survival probabilities, the predicted longitudinal trajectories differ between the packages. The trajectories in `JMbayes` have a much stronger increase in the log serum bilirubin level. Note, that the credibility intervals cannot be compared since they are computed differently. The differences in the predicted longitudinal outcomes are probably due to the different modeling of the longitudinal predictors.

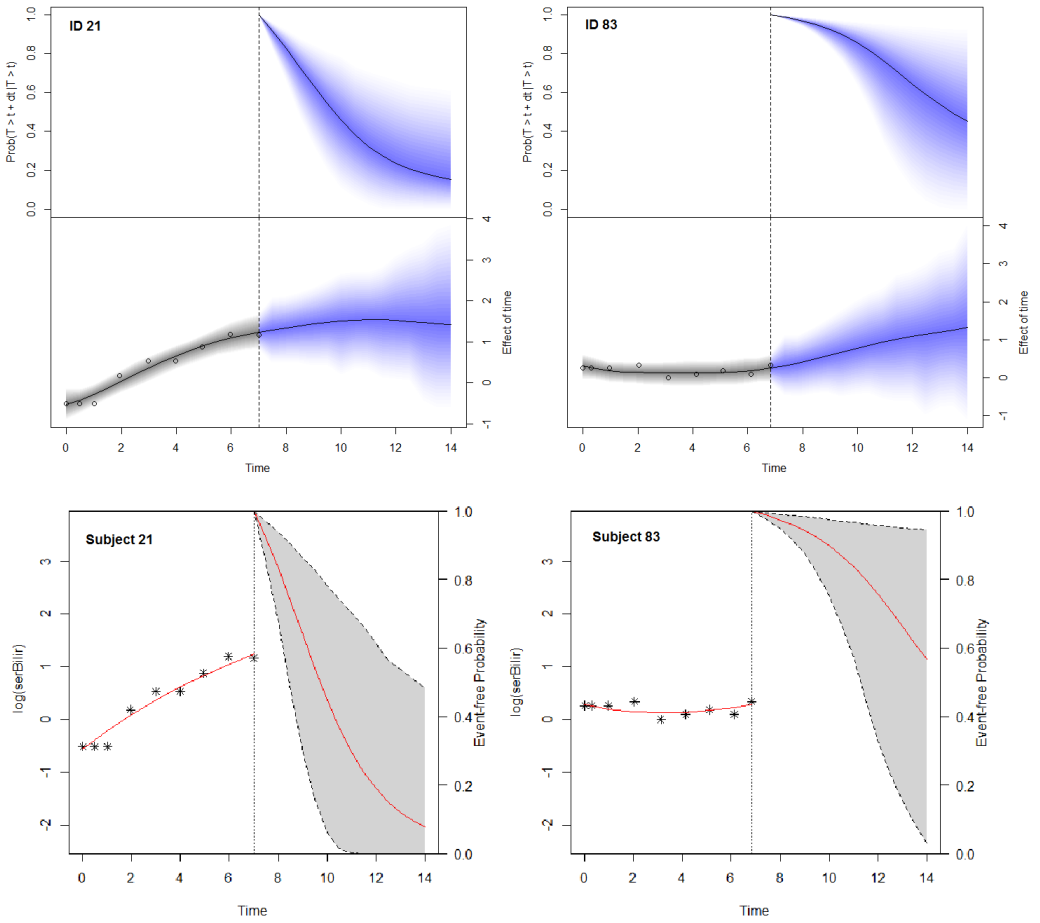

Figure 9: Predicted survival probabilities for Subjects 21 and 83 from PBC data estimated in `bamlss` (upper) and `JMbayes` (lower). Solid lines are mean estimates and area around are 95% credibility intervals. Dashed vertical line indicates last observation time point.

In order to further compare the predictive quality in both packages we evaluate the dynamic predictions by means of the dynamic C index and the integrated prediction error. To be better able

to compare both packages we model a constant association in both joint model fits. Therefore, we fit a new joint model in `bamlss` which is exactly modeled as described in Section 5.2 only assuming η_α to be constant. The results for all three model fits are presented in Table 5. Where columns three and four present the results for an in-sample evaluation and the results in columns five and six are based on a 10-fold cross validation. For the in-sample evaluation both models in `bamlss` yield a

		In-sample		*Cross-validation*	
Package	η_α	$\hat{C}^{\Delta 2}_{\text{dyn}}$	$\hat{\text{IPE}}(5,2)$	$\hat{C}^{\Delta 2}_{\text{dyn, 10-CV}}$	$\hat{\text{IPE}}_{\text{10-CV}}(5,2)$
`bamlss`	time-varying	0.8751	0.0486	0.8601	0.0712
		(23.8)	(42.7)	(2.1)	(0.5)
`bamlss`	constant	0.8794	0.0471	0.8644	0.0717
		(22.14)	(41.54)	(2.0)	(0.46)
`JMbayes`	constant	0.8611	0.0516	0.848	0.0530
		(7.2)	(14.9)	(0.41)	(0.31)

Table 5: Evaluation of the predictive quality in packages `bamlss` and `JMbayes` by means of the dynamic C index ($C^{\Delta 2}_{\text{dyn}}$) and the integrated prediction error (IPE$(5,2)$) using 100 Monte Carlo iterations. Both measures use a prediction interval of length 2 where the IPE uses all observations from the first 5 years. Numbers in parenthesis indicate computation time (in minutes) on a 3.40 GHz Intel Xeon Processor E5-2643.

higher dynamic C index and a lower integrated prediction error compared to `JMbayes`. For the cross validation `bamlss` yields a higher discrimination index but in terms of the integrated prediction error `JMbayes` outperforms `bamlss`. That is because `JMbayes` uses different time points for the computation of the prediction errors. In detail, the integrated prediction error is calculated as a weighted sum (4.24) of prediction errors where each prediction error is computed at the individual actual observed event times. In our implementation we use the actual observed event times in the new provided data which in this case are the testing data. However, in `JMbayes` those time points are taken from the data the model was fitted to (training data). Re-estimating the integrated prediction error for the joint model fit in `JMbayes` but using the same evaluation points as in `bamlss` we get $\hat{\text{IPE}}_{\text{10-CV}}(5,2) = 0.0755$ which is higher than both results in `bamlss`. Comparing both model fits in `bamlss` a constant modeling of predictor η_α results in almost the same evaluation measures. This is probably due to the fact that the time-varying smooth effect modeled in η_α rather indicates a constant association between marker and hazard which can be seen in Figure 4.

6 Simulation

We assess the predictive quality of the implemented dynamic prediction for flexible Bayesian additive joint models by means of a simulation study. For a time-constant η_α we are especially interested in comparing our results to the ones obtained from the package `JMbayes`. Moreover, we are going to evaluate the dynamic prediction for models where the true underlying association between the longitudinal marker and the time-to-event is time-varying. In those settings we aim to examine whether modeling a time-varying predictor η_α improves the performance of the dynamic prediction by comparing it to models that assume a constant predictor η_α. For both approaches, we are going to base the dynamic predictions on the joint models fitted in the simulation study by Köhler et al. (2017) where they use two different simulated data settings.

6.1 Simulation design

6.1.1 Data and Model

The difficulty when simulating survival data is that we are, in fact, not directly modeling the survival times but rather a hazard function. In this case the simulated variables cannot be directly connected to the survival times via the pre-specified coefficients as it is for example the case for linear regressions. Therefore, Bender et al. (2005) developed a method to compute survival times when simulating data for a Cox proportional hazard model. This method was extended by Crowther and Lambert (2013) which is suitable for more complex hazard models using numerical integration and root-finding algorithms.

The survival function for a Cox proportional hazard model is defined as

$$\begin{aligned} S_i(t) = \Pr(T_i^* > t) &= \exp\left[-\int_0^t h_0(u)\exp(\boldsymbol{w}_i^\top\boldsymbol{\gamma})\,du\right] \\ &= \exp[H_0(t)\exp(\boldsymbol{w}_i^\top\boldsymbol{\gamma})] \end{aligned} \tag{6.1}$$

with $H_0(t) = \int_0^t h_0(u)\,du$ denoting the integrated baseline hazard and $\boldsymbol{w}$ and $\boldsymbol{\gamma}$ being the observed time-constant covariates and the corresponding coefficient vector. The corresponding distribution function for the survival time T_i^* is then $F_i(t) = 1 - S_i(t)$. Using the facts that $F_i(T_i^*) = U$ with $U \sim \mathcal{U}(0,1)$ and $1 - U \sim \mathcal{U}(0,1)$, we get for the survival function

$$1 - U = S_i(T_i^*), \tag{6.2}$$

or equivalently

$$U = S_i(T_i^*) = \exp[H_0(T_i^*)\exp(\boldsymbol{w}_i^\top\boldsymbol{\gamma})]. \tag{6.3}$$

Solving this expression w.r.t the survival time T_i^* yields

$$T_i^* = H_0^{-1}\left[-\log(U)\exp(-\boldsymbol{w}_i^\top\boldsymbol{\gamma})\right], \tag{6.4}$$

where U is a random variable following a uniform distribution on the interval $[0,1]$. Therefore, when sampling a value for U the survival time T_i^* can be computed.

Not all baseline hazards can be easily inverted and integrated. Plus, in the case of joint models the covariates are time-varying. Therefore, Crowther and Lambert (2013) suggest finding values for T_i^* such that

$$S_i(T_i^*) - U = 0. \tag{6.5}$$

This is done by sampling a value for U from $\mathcal{U}(0,1)$ and then finding the roots using for example a Newton-Raphson algorithm. The integral for the cumulative hazard is approximated using Gaussian quadrature.

Following the simulation design of Köhler et al. (2017), every data setting consists of n subjects that are followed over a sequence of fixed time points $\mathcal{P}$. The true longitudinal marker $\eta_\mu(t)$ is calculated as in (3.16) with

- the effect of time $f_{\mu 1}(t) = 0.1(t+2)\exp(-0.075t)$,
- random subject-specific intercepts $f_{\mu 2}(i) = r_i$ where $r_i \sim \mathcal{N}(0, 0.025)$,
- functional random intercepts $f_{\mu 3}(t,i) = \boldsymbol{X}_{\mu 3}\boldsymbol{\beta}_{\mu 3}$,

- a global intercept $f_{\mu 4}(\boldsymbol{x}_{\mu i}) = 0.5$, and
- a covariate effect $f_{\mu 5} = 0.6 \sin(x_{2i})$ where $x_{2i} \sim \mathcal{U}(-3, 3)$.

The functional random intercepts are simulated using B-splines with 10 knots (two inner knots) and 6 basis functions, drawing the true vector of spline coefficients from $\boldsymbol{\beta}_{\mu 3} \sim \mathcal{N}(\mathbf{0}, (1/\tau_s^2)\tilde{\boldsymbol{K}}_s + (1/\tau_t^2)\tilde{\boldsymbol{K}}_t)$ as described in (3.18) with $\boldsymbol{K}_t = \boldsymbol{D}_2^\top \boldsymbol{D}_2$, $\tau_s^2 = 1$ and $\tau_t^2 = 0.2$. Köhler et al. (2017) compute the hazard function for each subject as stated in equation (3.12) with the true baseline hazard $\eta_\lambda(t) = 1.4 \log((t + 10)/1000)$ and baseline survival predictor $\eta_{\gamma i} = 0.5 \sin(x_{i1})$ with $x_{i1} \sim \mathcal{U}(-3, 3)$. The true association $\eta_\alpha(t)$ differs between the simulation settings assuming either a constant or time-varying effect. Being able to compute the hazard $h_i(t)$ the survival times are derived as described above. Every subject is censored after $\max(\mathcal{P})$ and additional uniform censoring is induced by drawing individual censoring times from $\mathcal{U}(0, 1.5 \cdot \max(\mathcal{P}))$. Moreover, the observed marker values y_{ij} are computed by adding independent errors $\varepsilon_{ij} \sim \mathcal{N}(0, 0.3^2)$ to each $\eta_{\mu i}(t_{ij})$. Missing data is generated by randomly discarding $p\%$ of the longitudinal observations from the original data. Using this general setup we simulate two different data settings: a and b. Data setting a consists of $n_a = 150$ subjects that are observed at time points $\mathcal{P}_a = [0, 1, 2, ..., 120]$ where $p_a = 75\%$ of the original data is randomly set to missing and on average 108 (72%) events occur. In b we have $n_b = 300$, $\mathcal{P}_b = [0, 3, 6, ..., 72]$ and $p_b = 10\%$ with on average 165 (55%) events leading to a larger data set compared to a.

Due to some changes in the function `simJM()` which is used for the generation of the data, we use the old `bamlss` package version `0.1-3` that is available in the supplementary material provided by Köhler et al. (2017) to generate the data. For more details on the changes of the function see Appendix D.1.

In total, Köhler et al. (2017) set up four different data and simulation schemes where for each scheme $Q = 200$ samples are drawn. In each sample q, for the computation of the posterior mean estimates, 23,000 MCMC samples are drawn including a burn-in phase of 3,000 and a thinning of 20 which results in 1,000 samples. The starting values for the MCMC sampling are always the corresponding posterior mode estimates.

The first two simulations, $1a$ and $1b$, assume $\eta_\alpha = 1$ in order to compare the results from the package `bamlss` to the ones obtained from the implementation of joint models in the package `JMBayes`.

The longitudinal predictor in each setting is modeled in two different ways. First we only include a random intercept and a random slope for the observation time points per subject. Then in the second fitting approach we include the aforementioned extension (3.16) which means modeling a random intercept, a smooth function of time and functional random intercepts. A general overview of both model set ups is presented in Table 6. Those two approaches of modeling the predictor η_μ lets us further explore if a more functional modeling is generally beneficial for the quality of the dynamic prediction.

For both modeling approaches the predictors η_λ and η_γ are modeled in `bamlss` using P-splines with cubic B-splines, second difference penalties and 10 knots (2 internal knots) which results in 5 basis functions per effect. In the less flexible model we model the longitudinal predictor η_μ with a smooth covariate effect using P-splines with 12 knots (4 internal knots) yielding 7 basis functions and a linear effect of time. For the random effects we only include subject-specific intercepts and a random slope. In the more flexible model η_μ is modeled as described in (3.16). We use P-splines with cubic B-splines, second difference penalties and 12 knots (4 internal knots) for the smooth effect of time, the functional random intercepts and the covariate effect yielding 7 basis functions

per effect and $7 \cdot n$ basis functions for the functional random intercepts after the application of the identifiability constraints.

In `JMbayes` the less flexible longitudinal model includes a smooth covariate effect using B-splines with no in internal knot in setting $1a$ and one internal knot in $1b$, a linear effect of time as well as a random intercept and slope per subject. The corresponding longitudinal model in the flexible approach is fitted using cubic B-splines for the fixed and random effects with one internal knot for the larger data setting b and no internal knot for setting a, yielding 4 and 3 basis functions. In both approaches the baseline survival predictor η_γ is modeled using cubic B-splines with 2 internal knots resulting in 5 basis functions. The baseline hazard is modeled using P-splines with default arguments in `JMbayes`, meaning cubic B-splines, second order difference penalties and 17 basis functions. The number of knots for both submodels in `JMbayes` were chosen such that the DIC is minimized. For the MCMC sampling the default arguments are used resulting in 20,000 iterations with a burn-in of 3,000 and thinning which leaves 2,000 MCMC samples.

In the second simulation settings, $2a$ and $2b$, where the true underlying association is nonlinear all predictors but η_α are specified as before. For data setting a this true association is $\eta_\alpha(t) = \cos((t-33)/33)$ and for data setting b $\eta_\alpha(t) = \cos(t-20)/20$ in order to achieve a similar shape in spite of the different time sequences. For both settings we fit two different models in `bamlss` in order to investigate if a time-varying modeling of $\eta_a lpha$ improves the dynamic prediction. First we model a time-varying association using P-splines with 10 knots (2 internal knots) which results in 5 basis functions when applying the constraints and second assuming a constant association. All other predictors are modeled as in setting 1.

Note that the joint model fit in `bamlss` can last from 10-14 hours (setting $1a$) up to 7-8 days (setting $2b$) per model highly depending on the simulation setting and the used processor. The estimation times in `JMbayes` are significantly lower ranging from 2-3 hours (setting $1a$) to 5-6 hours (setting $1b$). To reduce the total computation time the estimation of the joint models is parallelized where on each core one joint model is fitted.

A comparison of the resulting model fits in setting 1 by means of bias, MSE and coverage of the true parameter is given in the Appendix D.3. Note, that in setting $1a$ one model ($q = 91$) fails to converge in `JMbayes` which yields large biases and MSEs. Hence, we are going to exclude this model for the evaluation of the dynamic prediction.

6.1.2 Evaluation

The quality of our implemented dynamic prediction is going to be assessed based on the four introduced simulation schemes. The first two simulation settings, $1a$ and $1b$, are used to compare our results to the predictions obtained from the package `JMbayes`. Using the code and the old `bamlss` package version supplied by Köhler et al. (2017) in the corresponding supplementary material, we generate for each setting one new data set. For the smaller data set a we have $n_a = 500$ subjects and set $p_a = 30\%$ of the original data randomly to missing. For the larger data setting we follow $n_b = 700$ subjects where $p_b = 5\%$ of the original longitudinal observations are missing. This leads to 363 (73%) events and on average 28.7 observations per subject in setting a and 400 (57%) events in data set b with on average 9.5 longitudinal observations per subject. Each fitted joint model is then going to be evaluated by means of discrimination and calibration measures using the same new generated data set for all 200 models. To reduce the computation time, the estimation of the survival probabilities is based on the first-order estimator in all settings and both packages. In setting 1 for the computation of the dynamic C index $\mathrm{C}_{\mathrm{dyn}}^{\Delta t}$ we will use 6 different lengths for the prediction interval for $1a$, which are $\Delta t_a = \{3, 5, 10, 15, 20, 30\}$, and five different Δts for b, $\Delta t_b = \{3, 6, 12, 18, 24\}$, that are adjusted to the different time scale. Further the integrated prediction error is estimated

Predictor	flexible (`bamlss/JMbayes`)	less flexible
$\boldsymbol{\eta_\mu}$	- global intercept - subject-specific intercepts - smooth effect of time (*P-splines/B-splines*) - subject-specific deviations from global smooth time effect (*P-splines/B-splines*) - smooth covariate effect (*P-splines/B-splines*)	- global intercept - subject-specific intercepts - linear effect of time - random slope - smooth covariate effect
$\boldsymbol{\eta_\lambda}$	- baseline hazard (*P-splines / P-splines*)	
$\boldsymbol{\eta_\gamma}$	- smooth covariate effect (*P-splines/B-splines*)	
$\boldsymbol{\eta_\alpha}$	- constant (Setting 1) / time-varying (Setting 2)	
$\boldsymbol{\eta_\sigma}$	- constant	

Table 6: Simulation model setup. All terms for each predictor in flexible (left) and less flexible (right) approach. For the flexible model the entries in parenthesis indicate how smooth effects are modeled in `bamlss` (first) and `JMbayes` (second).

for several distinct intervals of same length such that almost the whole observation range is covered. That is for setting $1a$ we have 7 intervals of equal length 15, starting from 5 and ending at time point 110 ([5, 20], ..., [95, 110]). For data setting $1b$ we split the interval [6, 69] into seven intervals of length 9 ([6, 15], ..., [60, 69]). To reduce the computation times in setting 2 we only compute the dynamic C index for two different prediction interval lengths, one shorter and one longer interval ($\Delta t_a = \{5, 30\}$, $\Delta t_b = \{6, 24\}$). Moreover, we compare two different intervals for the integrated prediction error, one at the beginning and one rather in the end. For $2a$ we use the intervals [20, 35] and [80, 95] and for $2b$ the intervals [15, 24] and [60, 69].

The computation times highly differ between settings but also depend on the lengths of the prediction interval (dynamic C index) or the position of the interval (integrated prediction error). For instance on a 3.40 GHz Intel Xeon Processor E5-2643 the computation of the prediction error in the first interval for setting $1a$ takes around 5 hours in `bamlss` and 3 in `JMbayes` for each model. In setting $1b$ the computation takes on average 11 hours in `bamlss` and around 7 hours in `JMbayes`. The dynamic C index is estimated faster with on average 2 hours in `bamlss` and 40 to 60 minutes in `JMbayes` for setting $1a$. In setting $1b$ the computation takes around 4 hours (`bamlss`) or 2 hours (`JMbayes`) per model. To reduce the overall computation time we parallelized the computation where each core estimates the corresponding measure for one model.

Due to some minor bugs in the package `JMbayes` we used a slightly modified version which is attached in the supplementary material of this work. More details on the different modifications are given in the Appendix D.2. However, it is important to note that some modifications are only valid for the explicit setup of the simulation study described above and therefore this modified version can only be used in this context.

6.2 Simulation results

The main focus of this simulation study is the comparison of the predictive quality to `JMbayes`. For the discrimination measure (dynamic C index) the results for the models that include a more flexible modeling of the longitudinal trajectory are presented in Figure 10. The plots for the models that only model a random intercept and slope are shown in the Appendix D.4.

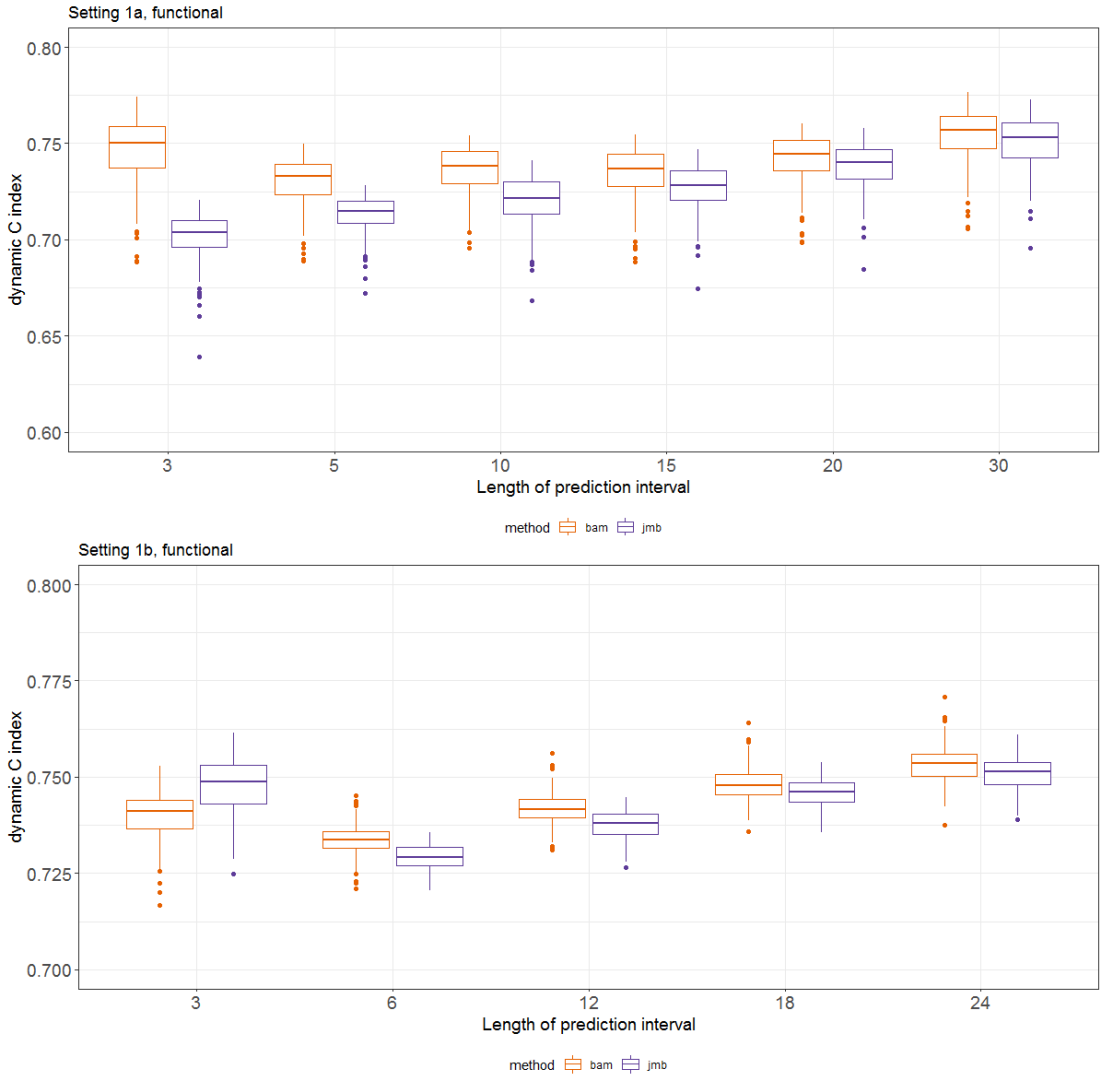

Figure 10: Dynamic C index (discrimination) for 200 simulated models that flexibly model the predictor η_μ evaluated based on one newly generated data set for prediction intervals of length {3, 5, 10, 15, 20, 30} in `bamlss` and `JMbayes`. The upper plot shows the results for data setting a and the lower for data setting b.

In Figure 10 we plot the results for the dynamic C index from both packages, `bamlss` in orange and `JMbayes` in blue, for several lengths of the prediction interval (x axis). Each boxplot represents the results for the 200 simulated models where each model was evaluated using the same newly generated data set. The upper plot shows the results for setting $1a$ and the lower plot setting $1b$. For both settings we generally see an increase of the boxplots for wider prediction intervals. This is the case since the computation of the dynamic C index is based on the weighted sum (4.21). In this sum we can always only consider the interval $[0, t_{\max} - \Delta t]$ where $t_{\max}$ is the last actual observation time point and Δt indicates the length of the prediction interval. Hence, for wider intervals we do not consider the time frame at the end where generally less people are still event free/uncensored and thus less events occur. We will illustrate this point again later in this Section.

In setting $1a$, for all different lengths of the prediction interval `bamlss` yields on average higher C indices than `JMbayes` which means our framework is better capable of discriminating between subjects that are going to have the event and those that do not have the event. This is particularly the case for shorter prediction intervals. For wider prediction intervals the C indices from both packages come closer but still `bamlss` performs better. For shorter prediction intervals (3 and 5) `bamlss` seems to scatter a little more. However, this becomes less for larger intervals such that both approaches have approximately the same variability. Further, comparing the results to the fit that only models a random slope and intercept in η_μ that is presented in the upper plot in Figure 24, we see that a more functional modeling of the longitudinal biomaker clearly improves the quality of the prediction in both packages.

The lower plot of Figure 10 shows the corresponding results for setting $1b$. On average `bamlss` has again a lower C index. Only for the shortest prediction interval which is of length 3 we see that `JMbayes` outperforms `bamlss`. As for setting a a more flexible modeling of the longitudinal predictor η_μ increases the overall performance of the dynamic prediction (compare to lower plot in Figure 24). Generally, it is striking that in the beginning there is a decrease in the C indices when widening the prediction interval. All 200 simulated models were only evaluated by means of one newly generated data set, therefore it is possible that the first decrease in the dynamic C indices is owed by the new data set. To verify this assumption we again evaluate the 200 models in both packages where this time each model is evaluated based on a different data set which is done by generating 200 new data sets. The outcome is shown in Figure 11 where now `bamlss` has on average a higher dynamic C index than `JMbayes` for all different lengths of the prediction interval and we further see a clear decrease in the C indices. Note that the generally higher variability in this approach is caused by the fact that we now have 200 different new data sets instead of only one.

As already pointed out before, the C indices generally increase when widening the prediction interval which we assume is connected to the actual interval used for the computation and thereby the number of the observed events. This idea is exemplified in Figure 12 where we kind of zoom into the computation of the dynamic C index in data setting a for a prediction interval of length 15. In the plot the points indicate the different medians of the AUCs at the time points that are used in the weighted sum (4.21). After time point 60 there is a clear decrease in the overall level of the AUCs. Comparing this to the distribution of the true event times in the data set used for the evaluation that is shown in the Appendix Figure 26 we see that there is as well a clear decrease in the actual observed events. Note, that in our framework we use a Gauss-Legendre quadrature for the integration of the dynamic C index whereas in `JMbayes` a Gauss-Kronrod quadrature is used. That is why we get slightly different evaluation time points in Figure 12.

The results for the integrated prediction error in the more flexible joint model fit are presented in Figure 13. The output for the less flexible models can be found in the Appendix Figure 25.

Both plots in Figure 13 show the results for the integrated prediction error that is computed at

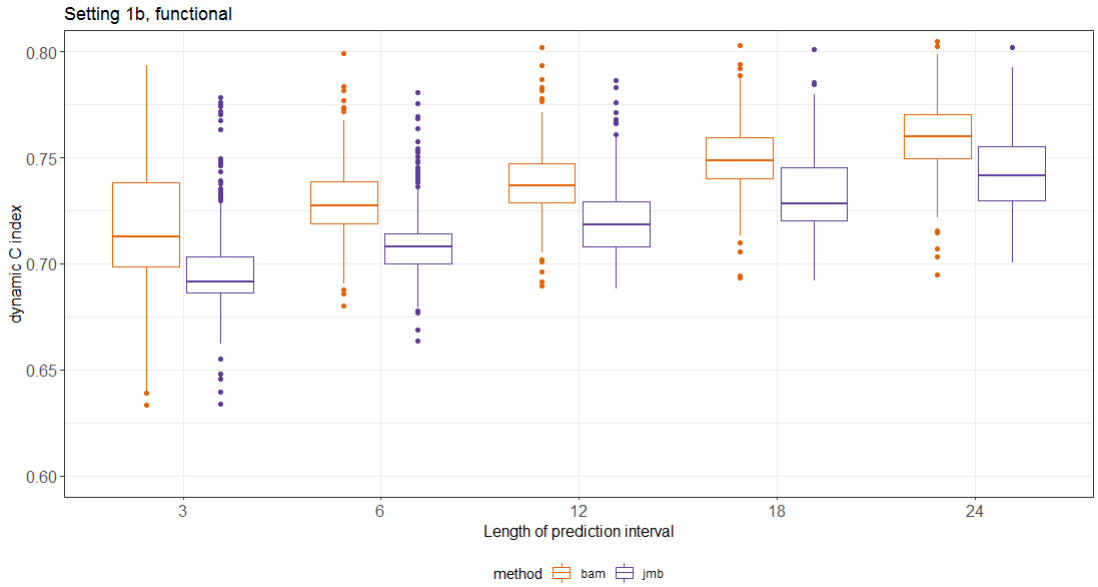

Figure 11: Dynamic C index (discrimination) for 200 simulated models in data setting b that flexibly model the predictor η_μ. Models are evaluated based on 200 newly generated data sets ($n_b = 300, p_b = 10\%$) for prediction intervals of length {3, 5, 10, 15, 20, 30} in `bamlss` (bam) and `JMbayes` (jmb).

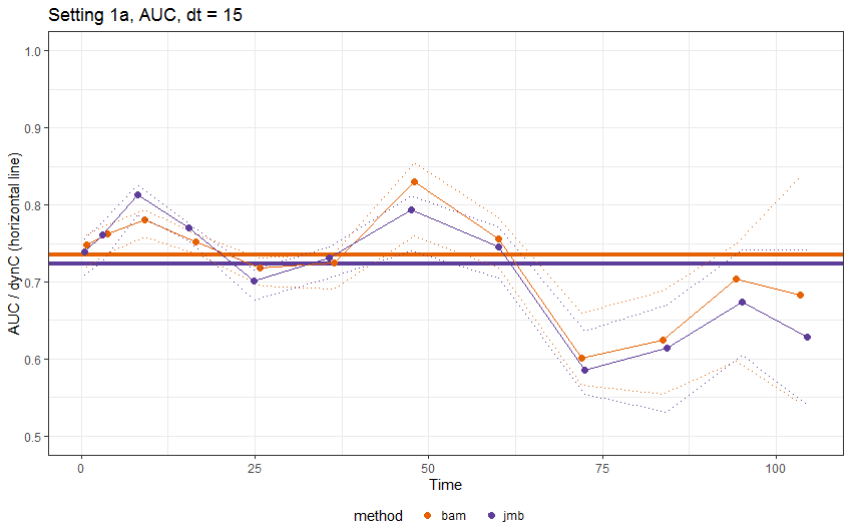

Figure 12: AUCs that are used for the computation of dynamic C index with a prediction interval of length 15. Time points at x axis indicate the cut off point for the prediction. Points are median AUC estimates, dotted lines present the 2.5% and 97.5% quantiles of the 200 models for `bamlss` (orange) and `JMbayes` (blue). The horizontal lines are the corresponding median estimates for the dynamic C index.

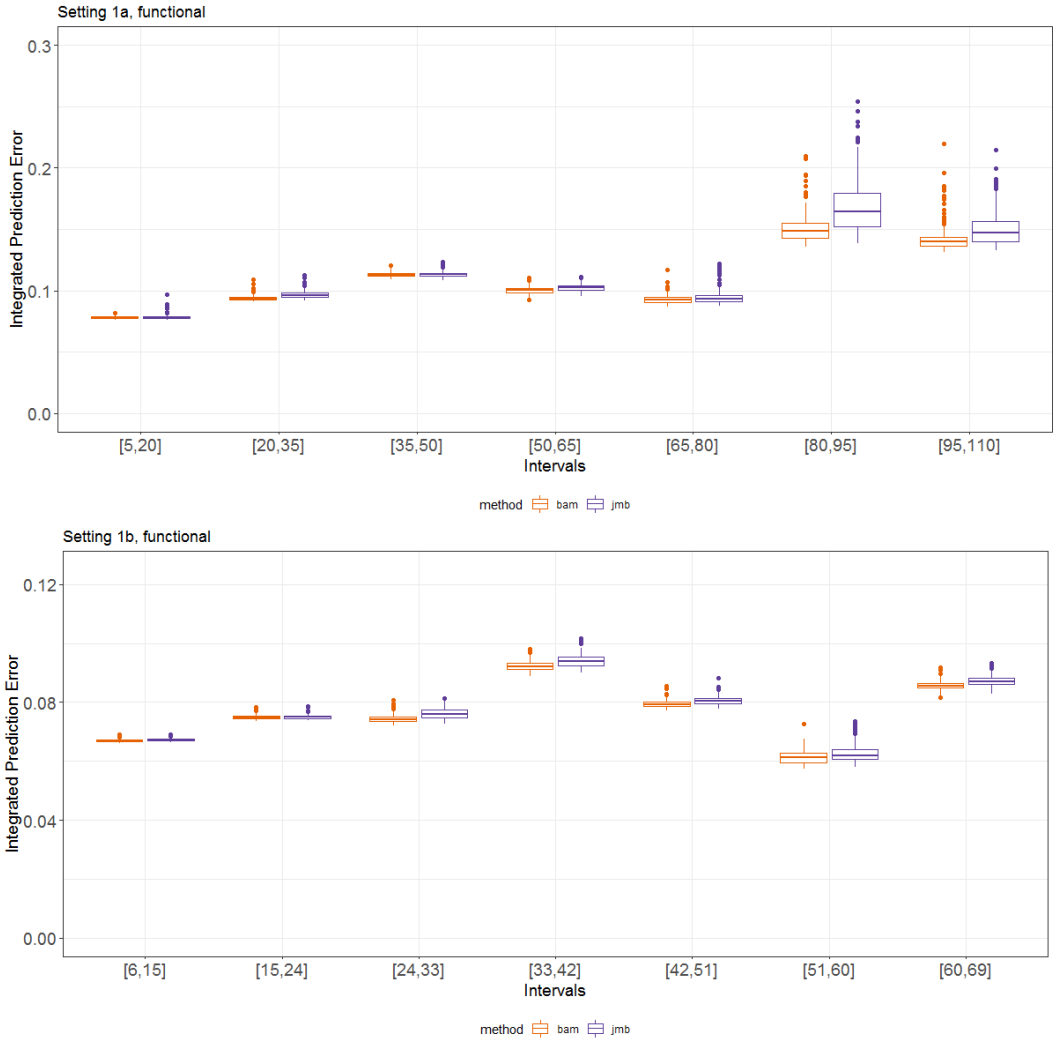

Figure 13: Integrated prediction error (calibration) for 200 simulated models that flexibly model the predictor η_μ evaluated for prediction intervals of same length at several distinct intervals in `bamlss` (bam) and `JMbayes` (jmb). The upper plot shows the results for data setting a and the lower for data setting b.

several distinct intervals each of length 15 for setting a and 9 for setting b. The boxplots present the results for all 200 models that are evaluated in each interval based on the same newly generated data set. Looking at the general course of the integrated prediction errors in both plots we cannot see a clear trend. Especially in setting b we rather see ups and downs for the whole range. For the first five intervals in data setting a (upper plot) the prediction errors for the two packages are approximately at the same level whereas the median estimates for `bamlss` are slightly lower compared to `JMbayes`. In the last two intervals the general level of the integrated prediction errors increase and they become more variable but still `bamlss` has a smaller prediction error on average. For data setting b we again see that both packages yield similar results but still the median estimates in `bamlss` exhibit smaller integrated prediction errors for all intervals. Comparing the results to the output for the less flexible joint models presented in Figure 25 we only see a slight improvement in some intervals when using a functional modeling for predictor η_μ.

As already indicated we do not see a clear trend for the integrated prediction errors. Those jumps are closely connected to the ratio of the number of actual observed events in the interval to the number of event free subjects at the end of the interval. Our simulation study suggests that for small intervals the dynamic prediction in both packages reveals larger errors when predicting actual events compared to predicting no event. For large intervals it is the other way around: the dynamic prediction in general yields smaller errors when predicting actual events than no events. In both settings compared to the whole range of observations we consider rather small prediction intervals. Thus, we get larger integrated prediction errors in intervals where we have a smaller ratio. For instance in setting b looking at interval [51, 60] where we have a relatively small error 15 events occurred in the interval and 89 subjects were still event free after time point 80 ($\frac{89}{15} = 5.93$) whereas in interval [42, 51] we observe 30 events and 117 subjects were event free ($\frac{117}{30} = 3.9$).

This theory is again illustrated in Figure 14 where we evaluate all 200 models fitted in setting a based on the one newly generated data set by means of the prediction error. We use all available information until time point 30 and then estimate the prediction error for several interval lengths. The lower plot presents the ratio of the number of subjects that are still event free by the end of the interval to the number of actual events occurring in the interval. This ratio is decreasing since the further we progress in time (which implies larger prediction intervals) the more actual events are observed in the interval and the less subjects are event free at the end. Therefore, in the beginning when having a large ratio for a short prediction interval (meaning larger prediction errors for actual events) we get small prediction errors. Then when preceding in time the number of events in the interval increases. Therefore the prediction errors increase as well until the point where the prediction interval is large enough such that the dynamic prediction yields smaller prediction errors when predicting actual events and larger when predicting no events. Since for those large prediction intervals the proportion of actual events in the interval is relatively high the prediction errors start to decrease again. Moreover, when comparing the performances of both packages, `bamlss` almost always outperforms `JMbayes` which matches the results from the integrated prediction error.

In simulation setting 2 we investigate if modeling a time-varying association between marker and event improves the quality of the dynamic prediction when the true underlying association is nonlinear. Therefore, we compare the resulting dynamic C indices and integrated prediction errors from joint models that model a time-varying predictor η_α to models only including a constant association. The output for the dynamic C index is presented in Figure 15 that indicates that the predictive quality generally profits from a time-varying modeling if the true underlying association is nonlinear. This is the case for shorter but also longer prediction intervals. Matching to our previous findings we again see an increase in the overall level for a longer prediction interval.

Figure 16 presents the results for the integrated prediction error. As before for the dynamic C

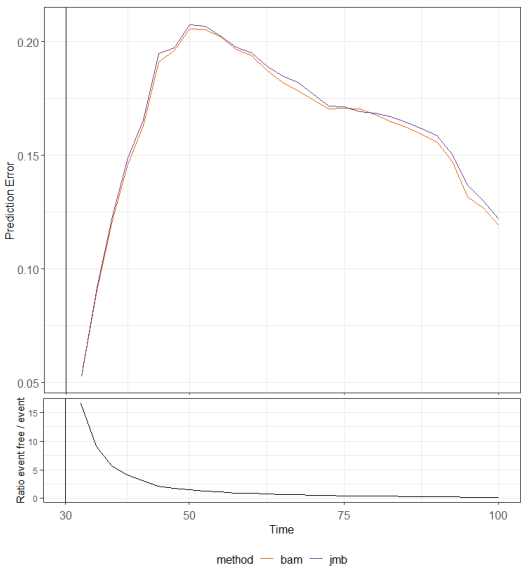

Figure 14: Prediction errors for several lengths of the prediction interval starting from 2.5 to 70 using all available information until time point 30. All 200 simulated models in setting a are evaluated based on one new generated data set ($n_a = 500$, $p_a = 0.3$). The solid lines show the corresponding mean estimates for `bamlss` (orange) and `JMbayes` (blue). Lower plot presents ratio of number of subjects that were event free at end of interval to number of actual observed events in the interval.

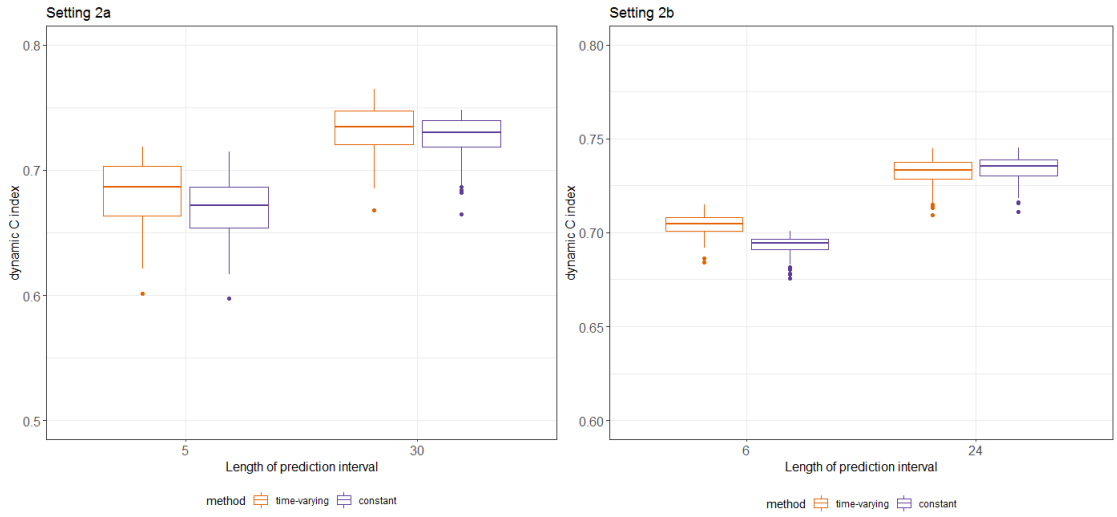

Figure 15: Dynamic C index for 200 simulated models where the true association between marker and event is nonlinear evaluated based on one newly generated data set per setting. Orange represents the results for a time varying and blue a constant predictor η_α in the joint model fit.

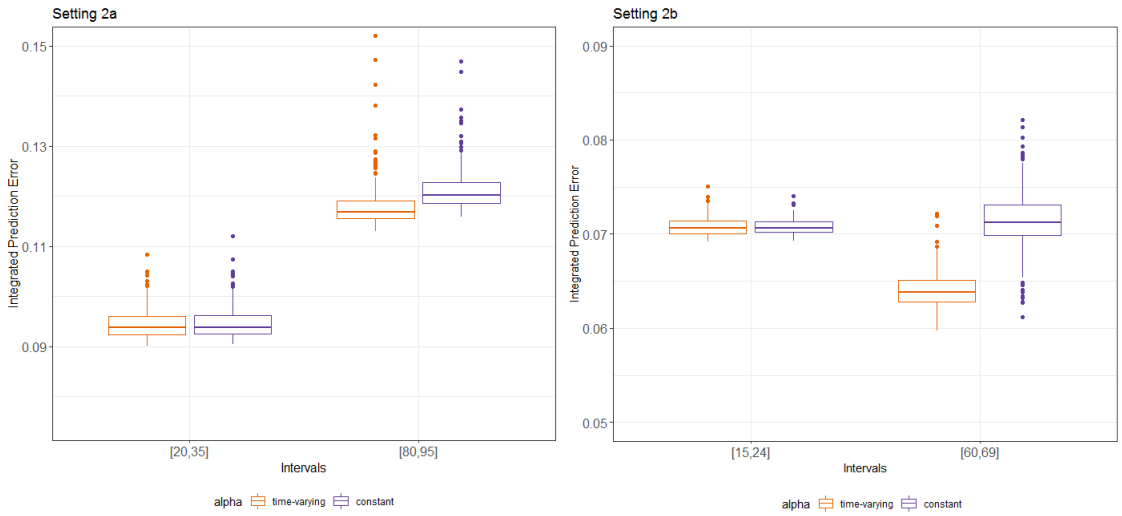

Figure 16: Integrated prediction error for 200 simulated models where the true association between marker and event is nonlinear evaluated based on one newly generated data set per setting. Orange represents the results for a time varying and blue a constant predictor η_α in the joint model fit.

index, modeling a time-varying association yields at least equally well results compared to a constant association. Analogous to our previous findings there is no clear trend in the overall course of the integrated prediction error.

In conclusion, in our simulation study we could show that our implemented dynamic prediction for flexible Bayesian additive joint models fitted in `bamlss` performs well in predicting the probability for future events, as well as in discriminating between subjects that are having the event and subjects that are still event free. Moreover, we could show that `bamlss` almost always outperforms the dynamic predictions in `JMbayes` when comparing the calibration and discrimination measures. Modeling the longitudinal predictor η_μ more flexible is especially beneficial when the aim is to discriminate between subjects. If the true underlying association between marker and event is nonlinear, the dynamic prediction can be generally improved by modeling a time-varying predictor η_α.

7 Discussion and Outlook

The core of the presented thesis was to extend the existing R-package `bamlss` by implementing a dynamic prediction for flexible Bayesian additive joint models and to further examine if the additional flexibility improves the predictive quality. For the dynamic prediction we proposed two different estimators: a first-order estimator that is directly based on the empirical Bayes estimates and the coefficient estimates obtained from the joint model fit; and an estimator based on a Monte Carlo simulation that also provides valid credibility intervals. The predictive quality can be assessed by means of discrimination and calibration measures that were adapted to our time dynamic setting. More specifically, for the discrimination we supply estimates for the AUC and a dynamic C index. For the calibration the prediction error as well as the integrated prediction error can be computed. The

framework of flexible Bayesian additive joint models offers a wide range of joint model specifications including the possibility to model structured additive predictors as well as a time-varying association between marker and event. Our implemented dynamic prediction is compatible with all mentioned joint model specifications.

We performed a re-analysis of the effect of the biomarker serum bilirubin on the liver disease primary biliary cirrhosis and thereby demonstrated the use of the implemented functions. Previous studies have already shown that there is a strong relation between the marker and the risk for a transplant or death (see for example Rizopoulos (2016); Köhler et al. (2018)). We could show that our prediction framework is well capable of predicting survival probabilities as well as discriminating between patients that are going to die/have a transplant within a time frame and patients that are still alive. Those results were further substantiated by a cross validation. Moreover, we compared the first-order estimator to the Monte Carlo simulation scheme. From that, we recommend using the first-order estimator if there is no need in obtaining credibility intervals since both approaches yield almost identical point estimates but the computation of survival probabilities in the Monte Carlo simulation is more time consuming.

To examine if the additional flexibility offered by the possible joint model specifications in `bamlss` is also beneficial for the dynamic prediction we compared our results to the ones obtained in `JMbayes` for a simliar model fit. Here, we could show that `bamlss` outperforms `JMbayes` not only in an in-sample evaluation but also in a cross-validation.

To further underpin our findings from the analysis of the PBC data we moreover conducted a simulation study. The aim of the simulation study was twofold. First, in settings $1a$ and $1b$, we were aiming to examine if a more flexible modeling of the longitudinal trajectory improves the predictive quality which was done comparing the dynamic predictions in our framework to the one implemented in package `JMbayes`, and second to investigate if a time-varying predictor η_α in the joint model fit improves the quality of the dynamic prediction. For the comparison of the predictive quality both packages were compared by means of the dynamic C index, a discrimination measure that takes the whole follow-up period into account, and the integrated prediction error which is a dynamic version of the prediction error. In almost all presented measures `bamlss` outperforms `JMbayes`. Another factor which should also be considered when thinking of real data application is the computational efficencency in terms of computation time. Here, `JMbayes` yields better results than `bamlss`. Moreover, we could show that a more flexible modeling of the longitudinal predictor η_μ in general improves the predictive quality in both packages, particularly when comparing the dynamic C indices. Further, if the true underlying association between marker and event is nonlinear, we could show that modeling a time-varying predictor η_α improves the overall dynamic prediction.

In summary, taking all presented results into account, we can conclude that generally the quality of the dynamic prediction can be improved by using flexible Bayesian additive joint models. Still both presented packages have its disadvantages. As already pointed out the joint model fit in `bamlss` is extremely time consuming compared to a similar fit in `JMbayes`. This could be avoided by basing the dynamic prediction in the `bamlss` framework on joint models fitted via posterior mode estimation which would significantly reduce the overall computation time. Köhler et al. (2017) have shown that the precision of posterior mode and mean estimates are generally similar. However, so far this approach does not yield any valid credible intervals in the dynamic prediction. Therefore, for future work it would be clearly beneficial for the dynamic prediction in `bamlss` to derive a method that is able to compute credible intervals for the first-order estimator which account for the variability from the coefficient estimates in the joint model fit and the empirical Bayes estimates. When dealing with real data the dynamic prediction in `JMbayes` might be too restrictive since it can only be used in the time range the model was fitted to. Moreover, both approaches assume independence of the

observation time points and the marker value which is violated in many observational studies since follow up visits are often scheduled based on the patient's condition. So far, to our knowledge, no dynamic prediction exists that can deal with this problem.

Within the presented dynamic prediction framework that is based on flexible Bayesian additive joint models, several extensions are possible. Another joint model specification in `bamlss` further allows to model a nonlinear association between marker and event (Köhler et al., 2018). Due to the increasing complexity we so far did not implement this specification for the dynamic prediction. Furthermore, the predictive quality outside of the observation range could be improved by implementing a prediction for P-splines as for example described in Currie et al. (2004). In this work, we mainly analyzed the quality of the predicted survival probabilities. For future work it could be an aim to also compare the predicted longitudinal outcomes of both presented packages by deriving appropriate validation measures.

A Technical details in Chapter 2.2.1

In the following we are going to proof the dependencies of the survival function, the hazard rate and the corresponding density of the true unobserved event time T^*. Therefore, we will start from the definition of the hazard function

$$h(t) = \lim_{dt \to 0} \frac{\Pr(t \leq T^* < t + dt \mid T^* \geq t)}{dt},$$

where the numerator can be reformulated as

$$\begin{aligned} \Pr(t \leq T^* < t + dt \mid T^* \geq t) &= \frac{\Pr(t \leq T^* < t + dt)}{\Pr(T^* \geq t)} \\ &= \frac{F(t + dt) - F(t)}{S(t)} \end{aligned}$$

with $F(t) = \Pr(T^* \leq t)$ being the distribution function of T^*. Plugging this back into the hazard we get

$$\begin{aligned} h(t) &= \frac{1}{S(t)} \lim_{dt \to 0} \frac{F(t + dt) - F(t)}{dt} \\ &= \frac{f(t)}{S(t)}, \end{aligned} \tag{A.1}$$

where the limit in the first equation is the definition of a derivative and thus equals the density $f(t)$. Having this representation of the hazard we can easily obtain the formula for the density $f(t) = h(t)S(t)$.

To show the dependency between hazard function and survival function let us start from the definition of the survival function $S(t) = 1 - F(t)$. Rearranging this expression and taking the derivative with respect to t yields

$$f(t) = -\frac{d\, S(t)}{d\, t}.$$

This means the density equals the negative derivative of the survival function. Combining this fact with equation (A.1) we can derive

$$h(t) = -\frac{d\, \ln\left(S(t)\right)}{d\, t}$$

since $\frac{d\, ln(S(t))}{d\, t} = -\frac{f(t)}{S(t)}$. Plugging this definition into the cumulative hazard rate and using the properties of the survival function we get

$$\Lambda(t) = -\ln\left(S(s)\right)\Big|_0^t = -\ln(S(t)) + \ln(\underbrace{S(0)}_{=1}) = -\ln(S(t)).$$

Thus, multiplying this equality by -1 and taking the exponential we will get $S(t) = \exp[-\Lambda(t)]$.

B Important algorithms

B.1 Gaussian quadrature rule

In general, a Gaussian quadrature rule approximates the value of an integral using a weighted sum of function evaluations, that is

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i),$$

with the nodes $-1 < x_1 < x_2 < ... < x_n < 1$ and positive weights w_i. The weights are chosen such that the above approximation is exact for the functions $f(x) = x^b$, $b = 0, 1, ..., n-1$. Therefore, we get the following linear system of equations for an even n

$$\begin{bmatrix} 1 & \dots & 1 \\ x_1 & \dots & x_n \\ \vdots & & \vdots \\ x_1^{n-1} & \dots & x_n^{n-1} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ \vdots \\ 2/n \end{bmatrix}. \tag{B.1}$$

The location of the nodes depends on the type of quadrature. The benefit of a Gaussian quadrature is that an n-point Gaussian quadrature is exact for all polynomials up to degree $n-1$. The nodes and its corresponding weights are typically defined on the interval $[-1, 1]$ but can be extended to the interval $[a, b]$ using the following transformation

$$\int_a^b f(x)dx \approx \frac{b-a}{2} \sum_{i=1}^{n} f\left(\frac{b-a}{2} x_i + \frac{a+b}{2}\right) w_i. \tag{B.2}$$

We get this expression by transforming the integral $\int_a^b f(x)$ to an integral on the interval $[-1, 1]$ (Schwarz and Köckler, 2011).

For the implementation of the dynamic prediction we specifically use the Gauss-Legendre quadrature. Here, the n nodes are chosen to equal the roots of the n-th Legendre polynomial on the interval $[-1, 1]$. In the following we are going to show that an n-point Gauss-Legendre quadrature exactly determines the integral for all polynomials up to degree $2n-1$ by following the proof in Chapter 7 in Schwarz and Köckler (2011). Therefore, we quickly introduce the system of Legendre polynomials which we denote by $\{L_n(x), n = 0, 1, 2, ...\}$. Legendre polynomials are orthogonal polynomials where the first n can be derived by applying the Gram-Schmidt orthogonalization to the functions $1, x, x^2, x^3, ..., x^n$. These polynomials have the following properties

1. $L_n(x)$ is of degree n
2. $L_n(x)$ has exactly n roots in the interval $[-1, 1]$
3. $\int_{-1}^{1} L_i(x)\, L_j(x)dx = 0$, if $i \neq j$, meaning the polynomials are orthogonal
4. $span(\{L_0(x), ..., L_n(x)\}) = span(\{1, x, ..., x^n\})$,

where the fourth property implies orthogonality between L_n and any polynomial up to degree $n-1$. Let $p(x)$ be a polynomial of degree $2n-1$ and $L_n(x)$ denote the n-th Legendre polynomial. Then, we get via polynomial division

$$p(x) : L_n(x) = q(x) + r(x),$$

where the quotient $q(x)$ is a polynomial of degree $\leq n-1$ and the remainder $r(x)$ with degree $\leq n-1$. This gives us the following representation

$$\int_{-1}^{1} p(x)dx = \underbrace{\int_{-1}^{1} q(x)L_n(x)dx}_{=0} + \int_{-1}^{1} r(x)dx,$$

where the second integral equals zero due to the fourth property of the Legendre polynomials. Moreover, we have the equalities

$$\begin{aligned}\sum_{i=1}^{n} w_i p(x_i) &= \sum_{i=1}^{n} w_i q(x_i) L_n(x_i) + \sum_{i=1}^{n} w_i r(x_i) = \sum_{i=1}^{n} w_i r(x_i) \\ &= \int_{-1}^{1} r(x_i) = \int_{-1}^{1} p(x_i)\end{aligned}$$

where the second equality holds since we are using the roots of L_n as our nodes. And the third equality is exact since we are using an n-point Gaussian quadrature for an $n-1$ polynomial with the above introduced weights w_i.

B.2 Newton-Raphson algorithm

The Newton-Raphson algorithm is a popular method to numerically find the root of a function that cannot be solved analytically. The idea behind this algorithm is to linearize the function at some point $x^{(k)}$ using a tangent. For this tangent we can then easily compute the root $x^{(k+1)}$ analytically which also serves as the starting point for the next iteration step. This procedure is then repeated until the algorithm converges. Formally, we get for the tangent by using a first order Taylor approximation

$$t(x) = f(x^{(k)}) + (x - x^{(k)})f'(x^{(k)}).$$

Setting $t(x) = 0$ and solving for x we get the iteration rule

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}. \tag{B.3}$$

For the estimation of parameters the Newton-Raphson algorithm is typically used to solve optimization problems such as the maximization of the log likelihood function. Hence we are searching for the roots of the derivative of a function f. In this case we can also interpret the Newton-Raphson algorithm in such way that we approximate f at some point $x^{(k)}$ using a second order Taylor approximation. Having a second order polynomial we can easily optimize this function analytically to get the starting values for the next iteration. In detail, we get for the second order polynomial

$$q(x) = f(x^{(k)}) + (x - x^{(k)})f'(x^{(k)}) + \frac{1}{2}(x - x^{(k)})^2 f''(x^{(k)}). \tag{B.4}$$

This quadratic function can be simply optimized by taking the first derivative

$$\frac{\partial q(x)}{\partial x} = f'(x^{(k)}) + (x - x^{(k)})f''(x^{(k)}). \tag{B.5}$$

and setting this expression equal to zero. Solving for x we get an equivalent iteration rule as above for the root of the first derivative of f.

B.3 Expectation Maximization algorithm

The Expectation Maximization (EM) algorithm is widely used to find maximum likelihood estimates in the case of missing data or latent variables. The general idea is to first estimate values for the latent variables (Expectation-step/E-step) and then using these values to optimize the likelihood (Maximization-step/M-step). To put this more formally, let $\boldsymbol{\vartheta}$ be the parameter vector and let $\boldsymbol{x}$ and $\boldsymbol{z}$ denote the observed and missing data, respectively, along with the observed data likelihood $L(\boldsymbol{\vartheta} \mid \boldsymbol{x})$ and the complete likelihood $L(\boldsymbol{\vartheta} \mid \boldsymbol{x}, \boldsymbol{z})$. The aim is to maximize the observed data likelihood. However, maximizing $L(\boldsymbol{\vartheta} \mid \boldsymbol{x})$ might be difficult. Instead, it is often easier to optimize the complete data likelihood. This yields the following algortihm

- Choose starting value $\boldsymbol{\vartheta}^{(0)}$
- Iterate $k = 1, 2, ...$ between E/M-steps until convergence of $\boldsymbol{\vartheta}^{(k)}$

  E-step: compute $Q(\boldsymbol{\vartheta} \mid \boldsymbol{\vartheta}^{(k)}) = E_{\boldsymbol{Z}|\boldsymbol{x},\boldsymbol{\vartheta}^{(k)}}[\log L(\boldsymbol{\vartheta} \mid \boldsymbol{x}, \boldsymbol{Z})]$
  M-step: compute $\boldsymbol{\vartheta}^{(k+1)} = \arg\max_{\boldsymbol{\vartheta}} Q(\boldsymbol{\vartheta} \mid \boldsymbol{\vartheta}^{(k)})$,

where the expectation in the **E**-step is taken with respect to $\boldsymbol{Z}$ given $\boldsymbol{x}$ and the current parameter values $\boldsymbol{\vartheta}^{(k)}$. Generally, it was shown that the EM algorithm converges rather slow compared to a Newton-Raphson algorithm. However, in each step the likelihood increases or at least does not decrease and guarantees convergence (Moon, 1996).

C PBC data

C.1 Diagnostics and Summary of model fit

We can check the convergence of the Markov chains by plotting the traceplots as well as the autocorrelation function via the method `plot` and setting the argument `which = "samples"`. Due to the limited space only the plots for the first 4 coefficients for β_α are shown in Figure 17 which indicates that the Markov chains well converged to the posterior distribution.

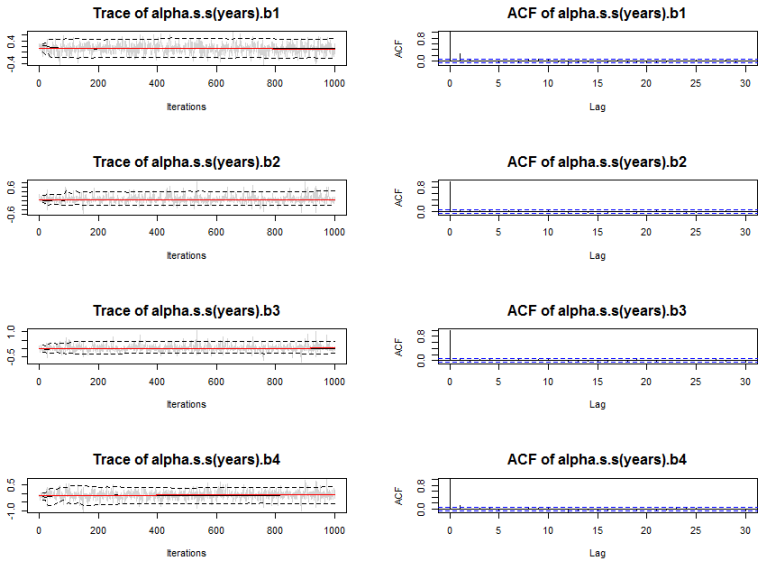

Figure 17: Traceplots (left) and autocorrelation functions (right) for first 4 estimated spline coefficients for predictor η_α.

To further examine the model results we present a plot for the effect of the smooth baseline hazard in Figure 18 that seems to be quite linear. The plots for the different terms in predictor η_μ are shown in Figure 19. In the left plot we see the subject-specific intercepts, in the middle the overall smooth time effect and in the right plot the functional random intercepts. The smooth time effect is increasing and relatively linear. Note that the wider credibility intervals for the smooth effects in the survival submodel compared to the credibility intervals in the longitudinal model are owed by the fact that we only use $n = 312$ observations for the survival model but $N = 1945$ observations in the longitudinal model.

C.2 Diagnostics of dynamic prediction

Figure 20 shows the traceplots and autocorrelation functions for the first four random effects coefficients of subject 21 that are obtained from a Metropolis-Hastings algorithm with 2000 iterations and a burnin-phase of 700 yielding in total 1300 samples. Comparing this output to the one obtained in Figure 7 that is based on a Metropolis-Hastings algorithm with 100 iterations and no burn-in phase we see that the traceplots become more constant and flat indicating a better convergence of

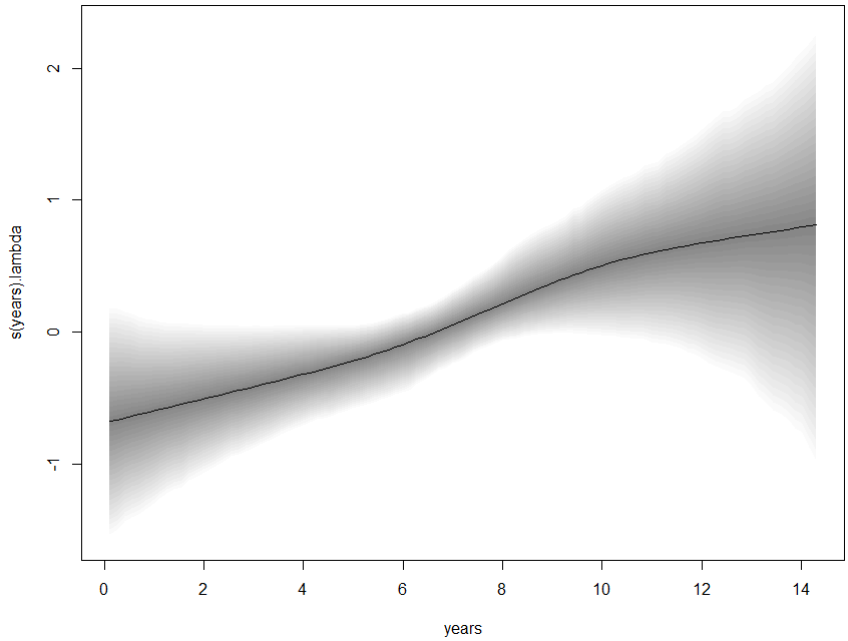

Figure 18: Point estimates (solid line) and credibility intervals (shaded area) for smooth effect of baseline hazard in the model fit for PBC data.

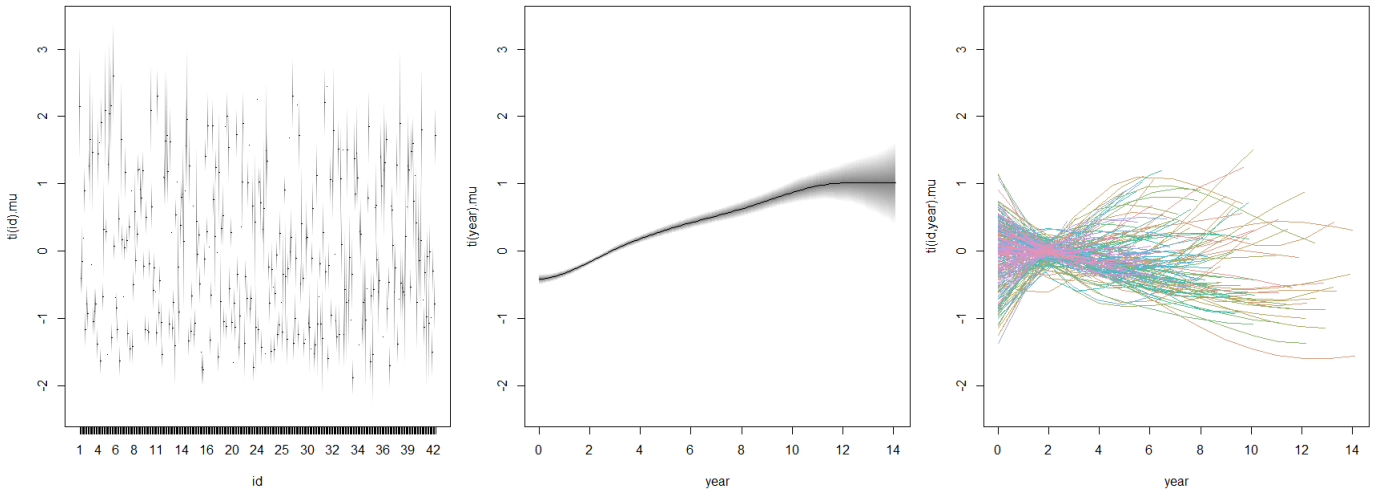

Figure 19: Effects for longitudinal predictor in model fit for PBC data including plots for random intercepts (left), smooth effect of time (middle) and functional random intercepts (right).

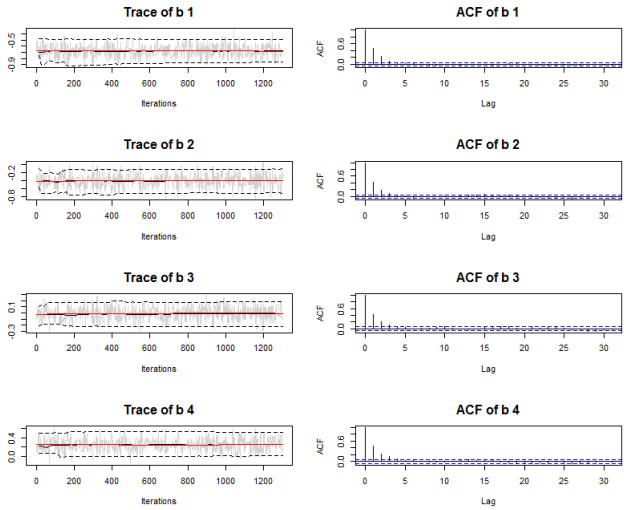

Figure 20: Traceplots (left) and autocorrelation functions (right) for first four random effects coefficients of subject 21 from PBC fit. Metropolis-Hastings algorithm is based on 2000 iterations including a burn-in of 700.

the chains. However, when comparing the dynamic predictions from those two approaches (Figures 5 and 21) we do not see any essential differences.

C.3 Comparison of packages

The smooth effect of time for the longitudinal submodel and the subject-specific deviations from this overall effect for the joint model fit in `JMbayes` are presented in Figure 22. Comparing those effects to the ones in `bamlss` (Figure 19) we see that the overall smooth time effect is more steep for `JMbayes` and further the individual curves for the deviations from this general effect are substantially different. Those differences are probably owed by the very different choices of basis functions.

The predicted longitudinal outcomes for patients 21 and 83 from the PBC data in `JMbayes` are presented in Figure 23. The dynamic predictions are obtained using the function `predict()`. Compared to the estimated smooth effect in `bamlss` the estimated effect in `JMbayes` is more steep and and further the subject-specific deviations are substentially different. We assume that those differences are due to the different modelings of the effects.

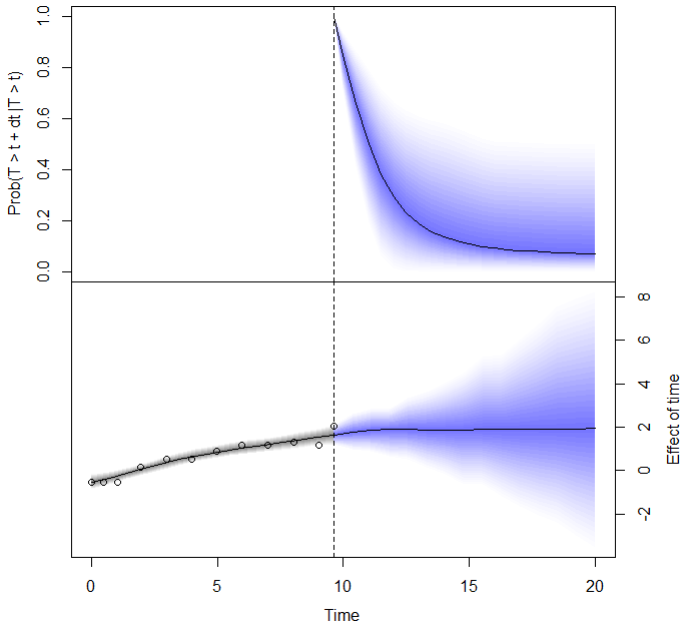

Figure 21: Dynamic prediction for patient 21 from PBC data that is based on a Metropolis-Hastings algorithm with 2000 iterations and a burn-in of 700 yielding 1300 samples.

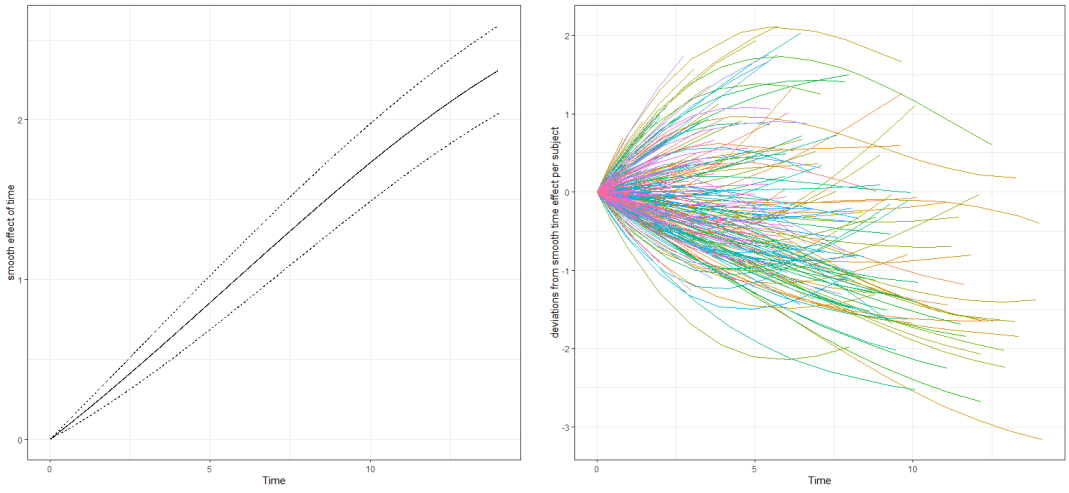

Figure 22: Effects for longitudinal predictor in `JMbayes` model fit for PBC data including smooth effect of time (left) and subject-specific deviations from this effect (right).

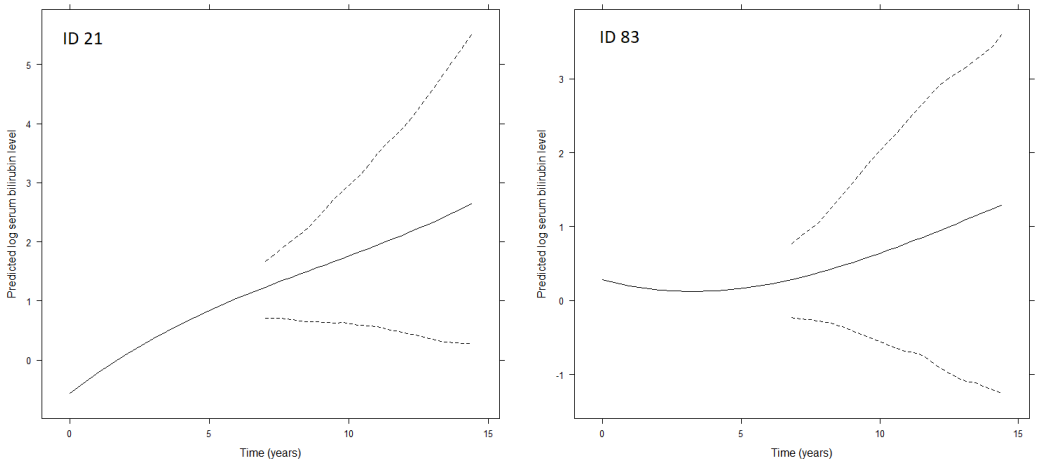

Figure 23: Predicted longitudinal outcomes for patients 21 and 83 from PBC data obtained from joint model fit in `JMbayes`. Solid lines present first-order estimates and dashed lines are 95% credibility intervals based in 100 Monte Carlo iterations.

D Simulation

D.1 Changes in data generating function

The package `bamlss` provides a function `simJM()` that allows to generate data for a joint model setup. However, due to changes in the calculation of the predictors η_λ and η_γ the old version `0.1-3` yields different data compared to the current package version `1.1-2` that is currently available at CRAN. Table 7 presents the changes between both versions.

Predictor	old version	new version
$\boldsymbol{\eta_\lambda}$	$1.4 \cdot \log(\texttt{time} + 10)/100$	$1.4 \cdot (\log(\texttt{time} + 10)/100) - 1.5$
$\boldsymbol{\eta_\gamma}$	$\sin(\texttt{x1})$	$0.3 \cdot \texttt{x1}$

Table 7: Changes in the computation for the predictors η_λ and η_γ in the function `simJM` between the `bamlss` package versions `0.1-3` (old version) and `1.1-2` (new version). The covariate `x1` is drawn from $\mathcal{U}(-3, 3)$ for each subject.

D.2 Modification JMbayes

In total three passages in the package `JMbayes` were changed in order to evaluate the simulations in Section 6.

First, the baseline survival predictor η_γ in the estimation of the joint models in `JMbayes` is modeled using cubic B-splines which are created using the function `pspline()` together with a roughness penalty close to zero, `theta = 0.001`. In the function for the dynamic prediction `survfitJM()` the design matrices for the new data are generated using the attributes of the resulting object rather than the direct function call. Hence, the default arguments are used for the function `psplines()` and the argument `theta` is disregarded which leads to an error. To overcome this problem, we extract the boundary knots from the attributes and pass the covariate, as well as those boundary

knots and the roughness penalty directly to the function `psplines()` in order to get the design matrix. Note, that this modification only works in the concrete simulation design and cannot be used for any other dynamic predictions.

Second, we fixed a bug in the function `S.b()` for the computation of the survival function that is especially used to calculate the dynamic predictions for the survival outcomes. If the survival function is to be computed at time point $t = 0$ the function returns $S(0) = 0$ although the value should be 1. We simply replaced the returned value by 1.

The last modification was in the function `prederrJM()` for the computation of the integrated prediction error (IPE) where the different prediction errors are computed at the time points of the actual events happening in this interval. In the package `JMbayes`, the actual event times for the interval are taken from the data the joint model was fitted to but not the provided new data for which the dynamic prediction should be evaluated. This can lead to errors if no event occurred in the original data but there are events happening in the new data set. We fixed this by taking the actual event times from the new data as the time points for the computation of the prediction error. Moreover, this modification makes the results for the integrated prediction errors from both packages comparable.

D.3 Model results

In the first simulation setting where we assume a constant association between event and marker our main focus is the comparison between the two packages `bamlss` and `JMbayes`. Therefore, Table 8 shows the bias, MSE and coverage of the true parameter for all predictors.

		aMSE		Bias		Coverage	
Predictor		1a	1b	1a	1b	1a	1b
η_α	`bamlss`	0.038	0.021	0.041	0.043	0.915	0.925
	`JMbayes`	17.35	0.021	0.389	0.048	0.850	0.880
$\eta_\gamma + \eta_\lambda$	`bamlss`	0.013	0.083	-0.0379	-0.038	0.932	0.943
	`JMbayes`	7468	0.101	-5.658	-0.048	0.744	0.742
η_μ	`bamlss`	0.025	0.031	0.0005	0.00002	0.942	0.946
	`JMbayes`	0.048	0.029	0.006	0.008	*	*

* No credibility intervals and thus no coverage could be calculated for these predictors.

Table 8: Model simulation results. Posterior mean estimation results from `bamlss` and `JMbayes` from data setting 1 (constant η_α) for small (a) and large (b) data sets.

When comparing the different measures it is striking that especially in Setting $1a$ the models fitted in `JMbayes` perform worse. Those poor values are especially owed by one single model ($q = 91$), that fails to properly fit the data. Particularly, the coefficient estimates for predictor η_γ are about 200 times larger compared to the other ones. Hence, Table 9 shows the same values again but this time leaving out model fit 91. When now comparing the results we see that both packages are relatively comparable. In both settings `bamlss` estimates the association η_α more precisely when comparing the bias and coverage. In general `bamlss` achieves a higher coverage for all predictors in both settings.

D.4 Further evaluation results

In this section we present the simulation results for the models that only use a random slope and intercept for modeling the predictor η_μ. The results for the discrimination measure are shown in

		aMSE		Bias		Coverage	
Predictor		1a	1b	1a	1b	1a	1b
η_α	`bamlss`	0.039	0.021	0.041	0.043	0.915	0.925
	`JMbayes`	0.047	0.021	0.095	0.049	0.854	0.879
$\eta_\gamma + \eta_\lambda$	`bamlss`	0.132	0.083	-0.038	-0.039	0.932	0.943
	`JMbayes`	0.151	0.101	-0.088	-0.049	0.748	0.743
η_μ	`bamlss`	0.025	0.031	0.0005	0.00001	0.942	0.946
	`JMbayes`	0.031	0.029	-0.001	0.008	*	*

* No credibility intervals and thus no coverage could be calculated for these predictors.

Table 9: Model simulation results without model 91. Posterior mean estimation results from `bamlss` and `JMbayes` from data setting 1 (constant η_α) for small (a) and large (b) data sets.

Figure 24 and the ones for the prediction error in Figure 25. For the dynamic C index we again see a general increasing trend for the overall C indices. Comparing the packages `bamlss` even outperform `JMbayes` when only modeling a less flexible longitudinal predictor. This is particularly the case for data setting b where `bamlss` not only exhibits on average higher C indices but also a smaller variability. Only for the smallest length of the prediction interval `JMbayes` performs better. However, we could show that this is owed by the single data set used for the evaluation.

In Figure 25 we present the integrated prediction errors for several distinct intervals where each interval has the same length. As for the more flexible models we do not see a clear trend but rather again those ups and downs in the same intervals. In general `bamlss` yields smaller integrated prediction errors but for all intervals both packages are approximately at the same level.

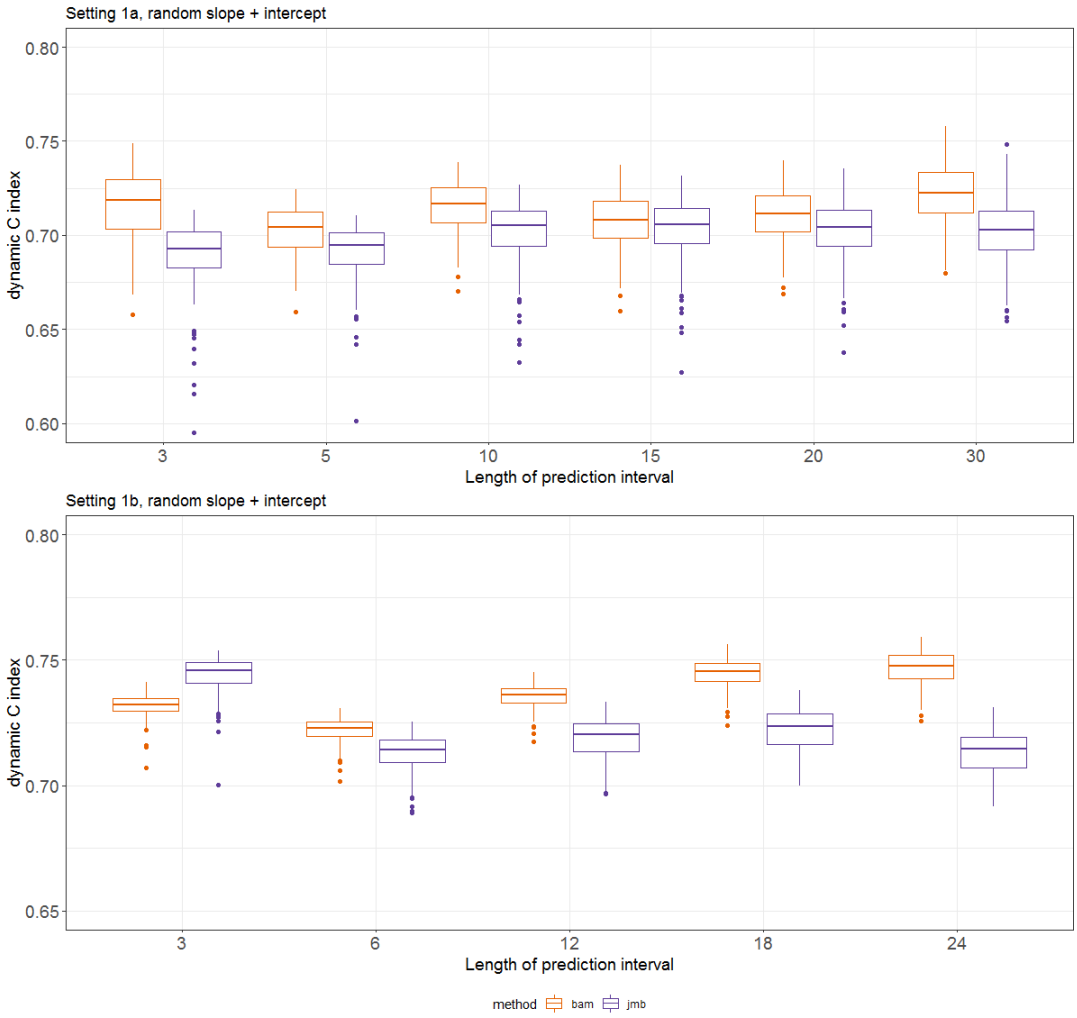

Figure 24: Caption

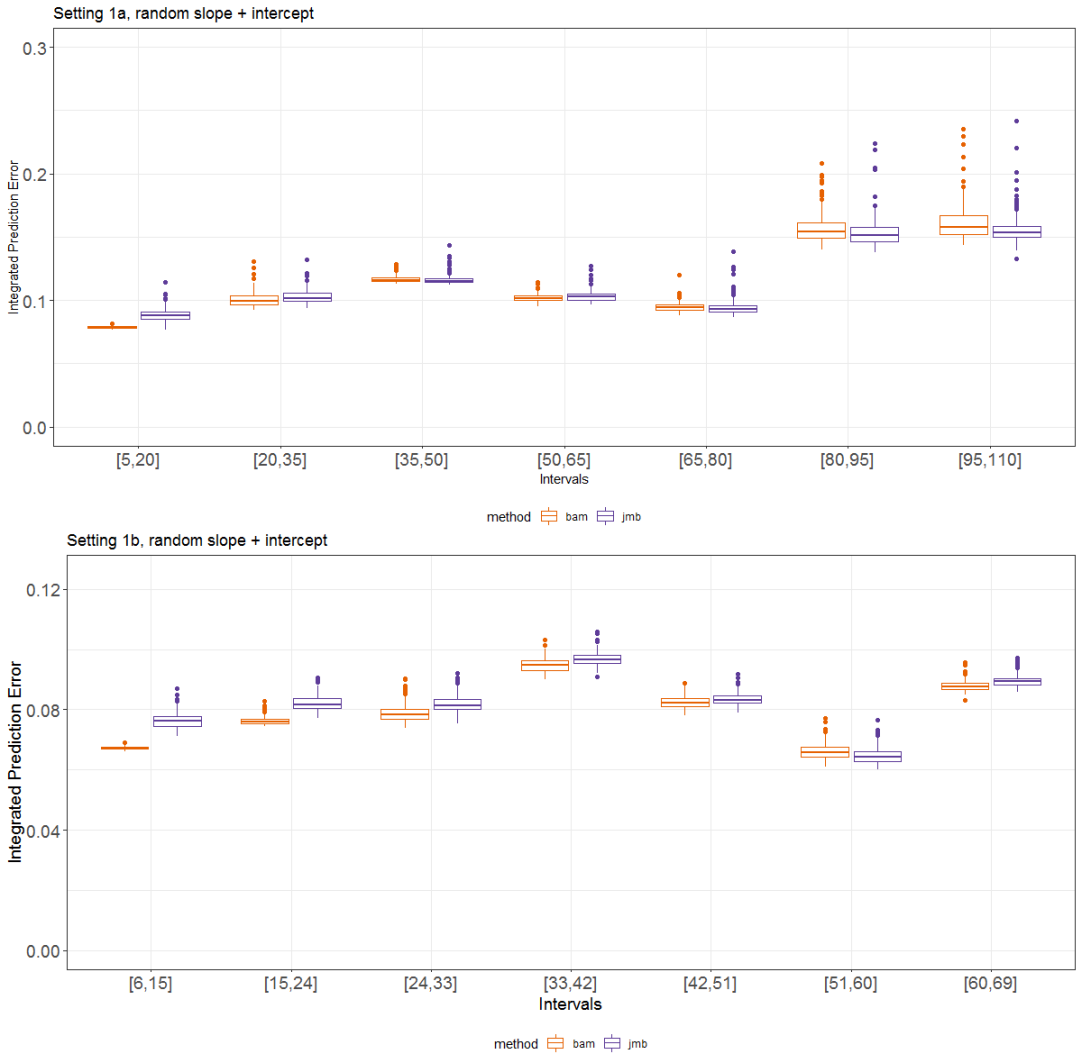

Figure 25: Caption

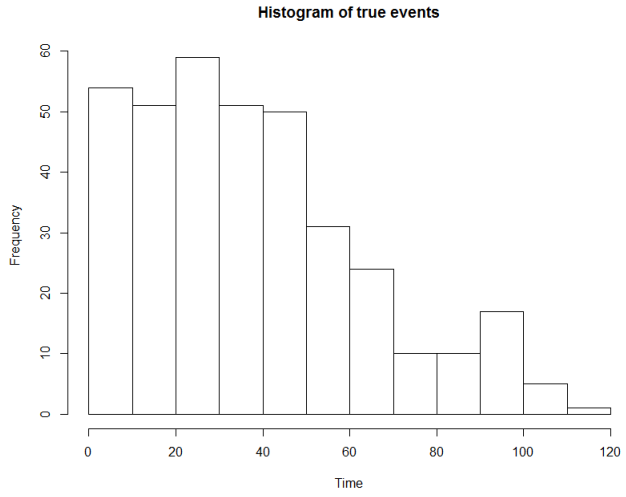

Figure 26: Histogram of the true observed events times in data set a that is used for the evaluation of the simulated models.

References

Andersen PK, Gill RD. Cox's Regression Model for Counting Processes: A Large Sample Study. The Annals of Statistics 1982;10(4):1100–1120. http://www.jstor.org/stable/2240714.

Andrinopoulou ER, Eilers PHC, Takkenberg JJM, Rizopoulos D. Improved dynamic predictions from joint models of longitudinal and survival data with time-varying effects using P-splines. Biometrics 2018;74(2):685–693. https://onlinelibrary.wiley.com/doi/abs/10.1111/biom.12814.

Antolini L, Boracchi P, Biganzoli E. A time-dependent discrimination index for survival data. Statistics in Medicine 2005;24(24):3927–3944. https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.2427.

Bender R, Augustin T, Blettner M. Generating survival times to simulate Cox proportional hazards models. Statistics in Medicine 2005;24(11):1713–1723. https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.2059.

Brown ER, Ibrahim JG, DeGruttola V. A Flexible B-Spline Model for Multiple Longitudinal Biomarkers and Survival. Biometrics 2005;61(1):64–73. https://onlinelibrary.wiley.com/doi/abs/10.1111/j.0006-341X.2005.030929.x.

Cox DR. Regression models and life-tables. Journal of the Royal Statistical Society: Series B (Methodological) 1972;34(2):187–202.

Crowther MJ, STJM: Stata module to fit shared parameter joint models of longitudinal and survival data; 2013. https://EconPapers.repec.org/RePEc:boc:bocode:s457502.

Crowther MJ, Lambert PC. Simulating biologically plausible complex survival data. Statistics in Medicine 2013;32(23):4118–4134. https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.5823.

Currie GP, Lee DK, Lipworth BJ. Long-Acting β 2-Agonists in Asthma. Drug safety 2006;29(8):647–656.

Currie ID, Durban M, Eilers PH. Smoothing and forecasting mortality rates. Statistical Modelling 2004;4(4):279–298. https://doi.org/10.1191/1471082X04st080oa.

Dafni UG, Tsiatis AA. Evaluating Surrogate Markers of Clinical Outcome When Measured with Error. Biometrics 1998;54(4):1445–1462. http://www.jstor.org/stable/2533670.

De Boor C. A practical guide to splines, vol. 27. Springer-Verlag New York; 1978.

Ding J, Wang JL. Modeling Longitudinal Data with Nonparametric Multiplicative Random Effects Jointly with Survival Data. Biometrics 2008;64(2):546–556. https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1541-0420.2007.00896.x.

Eilers PHC, Marx BD. Flexible smoothing with B -splines and penalties. Statist Sci 1996 05;11(2):89–121. https://doi.org/10.1214/ss/1038425655.

Fahrmeir L, Kneib T, Lang S. Regression - Modelle, Methoden und Anwendungen. 2. ed. Berlin Heidelberg New York: Springer-Verlag; 2009.

Fahrmeir L, Tutz G. Multivariate statistical modelling based on generalized linear models. Springer Science & Business Media; 2013.

Faucett CL, Thomas DC. Simultaneously modelling censored survival data and repeatedly measured covariates: A Gibbs sampling approach. Statistics in Medicine 1996;15(15):1663–1685. https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-0258%2819960815%2915%3A15%3C1663%3A%3AAID-SIM294%3E3.0.CO%3B2-1.

Fleming TR, Harrington DP. Counting processes and survival analysis, vol. 169. John Wiley & Sons; 2011.

Gould LA, Boye ME, Crowther MJ, Ibrahim JG, Quartey G, Micallef S, et al. Joint modeling of survival and longitudinal non-survival data: current methods and issues. Report of the DIA Bayesian joint modeling working group. Statistics in medicine 2015;34(14):2181–2195.

Heagerty PJ, Zheng Y. Survival Model Predictive Accuracy and ROC Curves. Biometrics 2005;61(1):92–105. https://onlinelibrary.wiley.com/doi/abs/10.1111/j.0006-341X.2005.030814.x.

Henderson R, Diggle P, Dobson A. Joint modelling of longitudinal measurements and event time data. Biostatistics 2000;1(4):465–480.

Henderson R, Diggle P, Dobson A. Identification and efficacy of longitudinal markers for survival. Biostatistics 2002 03;3(1):33–50. https://doi.org/10.1093/biostatistics/3.1.33.

Kalbfleisch JD, Prentice RL. The statistical analysis of failure time data, vol. 360. John Wiley & Sons; 2011.

Klein JP, Moeschberger ML. Survival analysis: techniques for censored and truncated data. Springer Science & Business Media; 2006.

Köhler M, Umlauf N, Beyerlein A, Winkler C, Ziegler AG, Greven S. Flexible Bayesian additive joint models with an application to type 1 diabetes research. Biometrical Journal 2017;59(6):1144–1165.

Köhler M, Umlauf N, Greven S. Nonlinear association structures in flexible Bayesian additive joint models. Statistics in Medicine 2018;37(30):4771–4788. https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.7967.

Lang S, Brezger A. Bayesian P-Splines. Journal of Computational and Graphical Statistics 2004;13(1):183–212. https://doi.org/10.1198/1061860043010.

Moon TK. The expectation-maximization algorithm. IEEE Signal Processing Magazine 1996;13(6):47–60.

Mukherjee D, Topol EJ. Pharmacogenomics in cardiovascular diseases. Progress in Cardiovascular Diseases 2002;44(6):479 – 498. http://www.sciencedirect.com/science/article/pii/S0033062002700212, unstable Plaque, Part II.

Papageorgiou G, Mauff K, Tomer A, Rizopoulos D. An overview of joint modeling of time-to-event and longitudinal outcomes. Annual review of statistics and its application 2019;.

Pawitan Y, Self S. Modeling Disease Marker Processes in AIDS. Journal of the American Statistical Association 1993;88(423):719–726. https://doi.org/10.1080/01621459.1993.10476332.

Pencina MJ, D' Agostino Sr RB, D' Agostino Jr RB, Vasan RS. Evaluating the added predictive ability of a new marker: From area under the ROC curve to reclassification and beyond. Statistics in Medicine 2008;27(2):157–172. https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.2929.

Philipson P, Sousa I, Diggle PJ, Williamson P. Package 'joineR' 2020;.

Prentice RL. Covariate Measurement Errors and Parameter Estimation in a Failure Time Regression Model. Biometrika 1982;69(2):331–342.

Proust-Lima C, Séne M, Taylor JM, Jacqmin-Gadda H. Joint latent class models for longitudinal and time-to-event data: A review. Statistical Methods in Medical Research 2014;23(1):74–90. https://doi.org/10.1177/0962280212445839, pMID: 22517270.

Proust-Lima C, Taylor JMG. Development and validation of a dynamic prognostic tool for prostate cancer recurrence using repeated measures of posttreatment PSA: a joint modeling approach. Biostatistics 2009 04;10(3):535–549. https://doi.org/10.1093/biostatistics/kxp009.

R Brown E, G Ibrahim J. A Bayesian Semiparametric Joint Hierarchical Model for Longitudinal and Survival Data. Biometrics 2003;59(2):221–228. https://onlinelibrary.wiley.com/doi/abs/10.1111/1541-0420.00028.

Rizopoulos D. JM: An R package for the joint modelling of longitudinal and time-to-event data. Journal of Statistical Software (Online) 2010;35(9):1–33.

Rizopoulos D. Dynamic predictions and prospective accuracy in joint models for longitudinal and time-to-event data. Biometrics 2011;67(3):819–829.

Rizopoulos D. Joint models for longitudinal and time-to-event data: With applications in R. Chapman and Hall/CRC; 2012.

Rizopoulos D. The R Package JMbayes for Fitting Joint Models for Longitudinal and Time-to-Event Data Using MCMC. Journal of Statistical Software, Articles 2016;72(7):1–46. https://www.jstatsoft.org/v072/i07.

Rizopoulos D, Ghosh P. A Bayesian semiparametric multivariate joint model for multiple longitudinal outcomes and a time-to-event. Statistics in Medicine 2011;30(12):1366–1380. https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.4205.

Rizopoulos D, Hatfield LA, Carlin BP, Takkenberg JJM. Combining Dynamic Predictions From Joint Models for Longitudinal and Time-to-Event Data Using Bayesian Model Averaging. Journal of the American Statistical Association 2014;109(508):1385–1397. https://doi.org/10.1080/01621459.2014.931236.

Rizopoulos D, Molenberghs G, Lesaffre EMEH. Dynamic predictions with time-dependent covariates in survival analysis using joint modeling and landmarking. Biometrical Journal 2017;59(6):1261–1276. https://onlinelibrary.wiley.com/doi/abs/10.1002/bimj.201600238.

Rizopoulos D, Taylor JMG, Van Rosmalen J, Steyerberg EW, Takkenberg JJM. Personalized screening intervals for biomarkers using joint models for longitudinal and survival data. Biostatistics 2015 08;17(1):149–164. https://doi.org/10.1093/biostatistics/kxv031.

Rizopoulos D, Verbeke G, Molenberghs G. Shared parameter models under random effects misspecification. Biometrika 2008 01;95(1):63–74. https://doi.org/10.1093/biomet/asm087.

Scheipl F, Staicu AM, Greven S. Functional Additive Mixed Models. Journal of Computational and Graphical Statistics 2015;24(2):477–501. https://doi.org/10.1080/10618600.2014.901914.

Schemper M, Henderson R. Predictive Accuracy and Explained Variation in Cox Regression. Biometrics 2000;56(1):249–255. https://onlinelibrary.wiley.com/doi/abs/10.1111/j.0006-341X.2000.00249.x.

Schwarz HR, Köckler N. Numerische Mathematik. Springer-Verlag; 2011.

Tomer A, Nieboer D, Roobol MJ, Steyerberg EW, Rizopoulos D. Personalized schedules for surveillance of low-risk prostate cancer patients. Biometrics 2019;75(1):153–162.

Tsiatis AA. A Large Sample Study of Cox's Regression Model. Ann Statist 1981 01;9(1):93–108. https://doi.org/10.1214/aos/1176345335.

Umlauf N, Klein N, Zeileis A. BAMLSS: Bayesian additive models for location, scale, and shape (and beyond). Journal of Computational and Graphical Statistics 2018;27(3):612–627.

Verbeke G, Molenberghs G. Linear Mixed Models for Longitudinal Data. Springer; 2000.

Wood SN. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 2011;73(1):3–36. https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2010.00749.x.

Wulfsohn MS, Tsiatis AA. A joint model for survival and longitudinal data measured with error. Biometrics 1997;p. 330–339.

Zhang D, Chen MH, Ibrahim JG, Boye ME, Shen W. JMFit: a SAS macro for joint models of longitudinal and survival data. Journal of statistical software 2016;71(3).

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

....

date (signature)